Jacobs Challenging today. Reinventing tomorrow.

AI-based Dynamic Arterial Signal Management – A Case Study

Zelan Jia, PE
Joyce Zhou, PE, PTOE
Shankar Natarajan, PE, PTOE

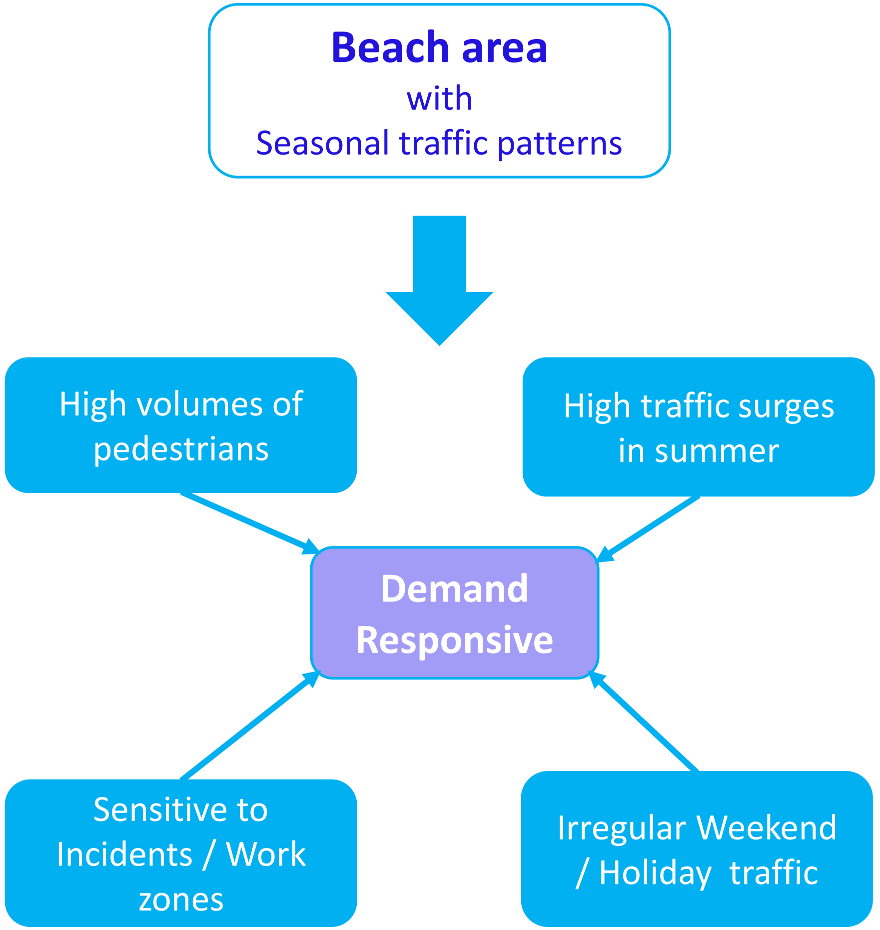
VASITE
June 2023

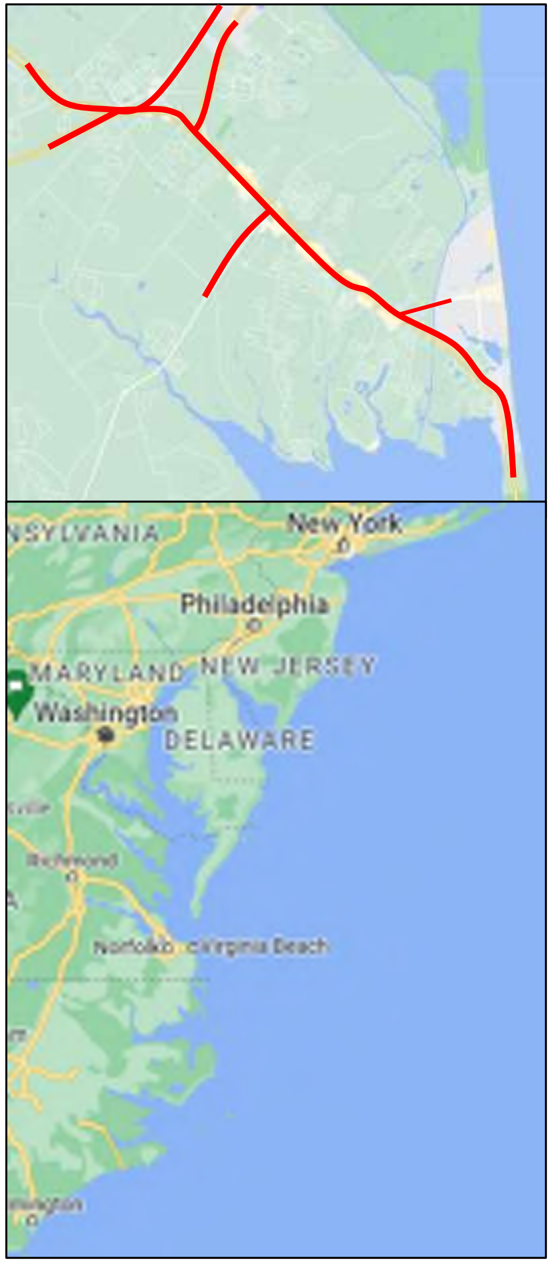
1. Introduction
2. Methodology
3. Traffic Volume & Occupancy Prediction
4. Signal Controller & API Development
5. Analysis Results
6. Conclusion

Introduction

Study Scope

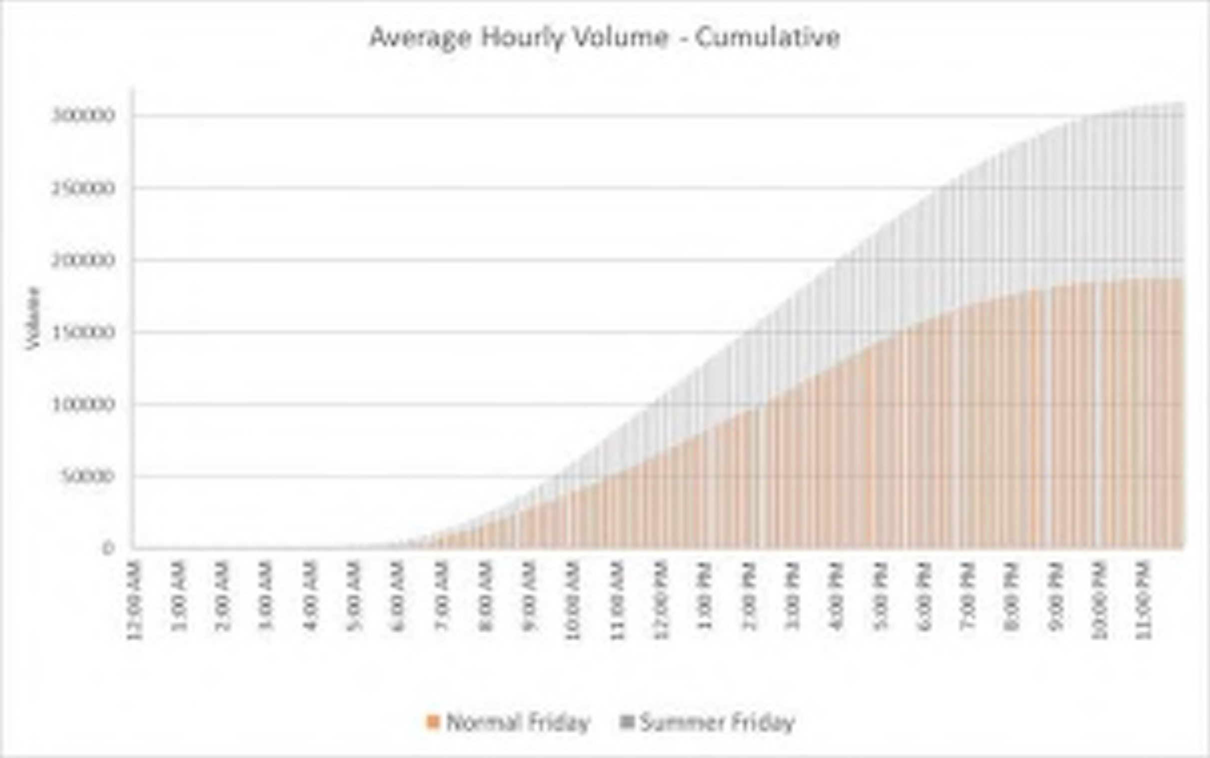
- Improve traffic operations on a beachbound arterial corridor
- Existing system is already actuated-coordinated with dynamic pattern changes based on real-time traffic conditions
- Traffic surges very fast during summer, reactive system cannot keep up
- Need for a proactive/ predictive system to manage pattern changes in real-time

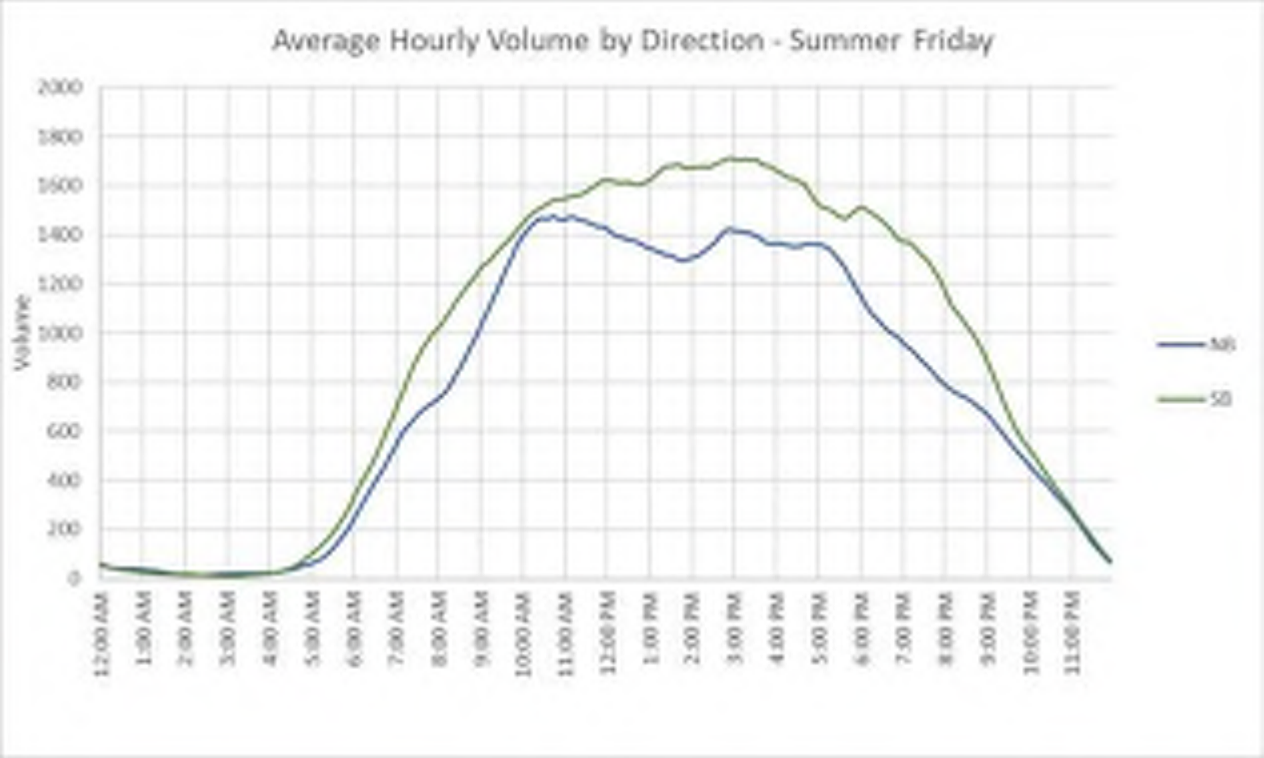




Study Scope

- High volume on weekends than weekdays
- Overall high volume daily on Friday (summertime)
- High southbound volume to beach area

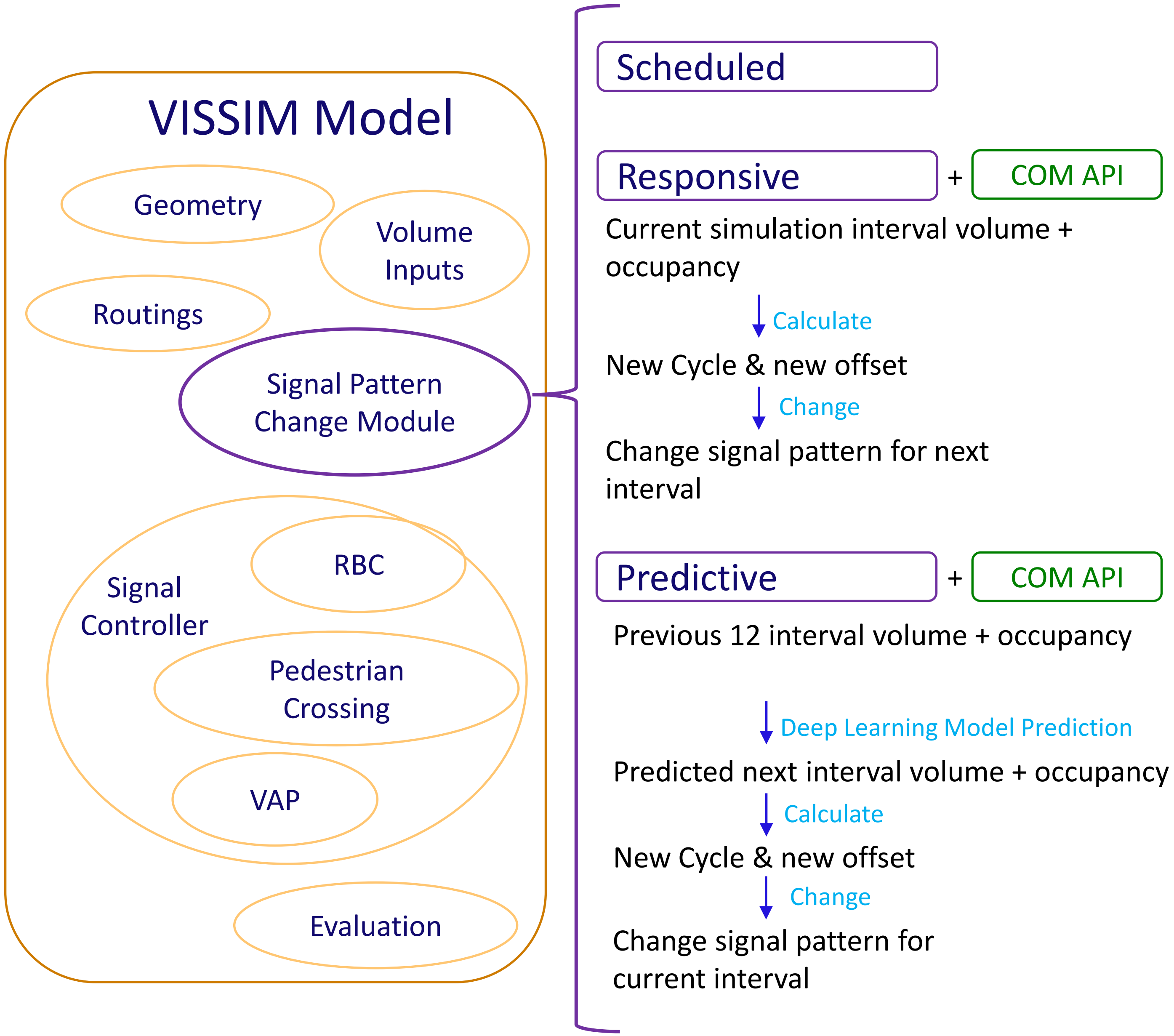




Methodology

Methodology

- Test alternative pattern change systems
  - Scheduled – Baseline
  - Responsive (Dynamic) – Current
  - Predictive - Planned
- Use Machine Learning (Deep Learning) algorithms for volume & occupancy prediction
  - Use predicted volume & occupancy to trigger responsive signal group signal cycle and offset change
- Model and evaluate system in microsimulation model
  - Collect data every 5-min simulation time interval.
- Compare with field data



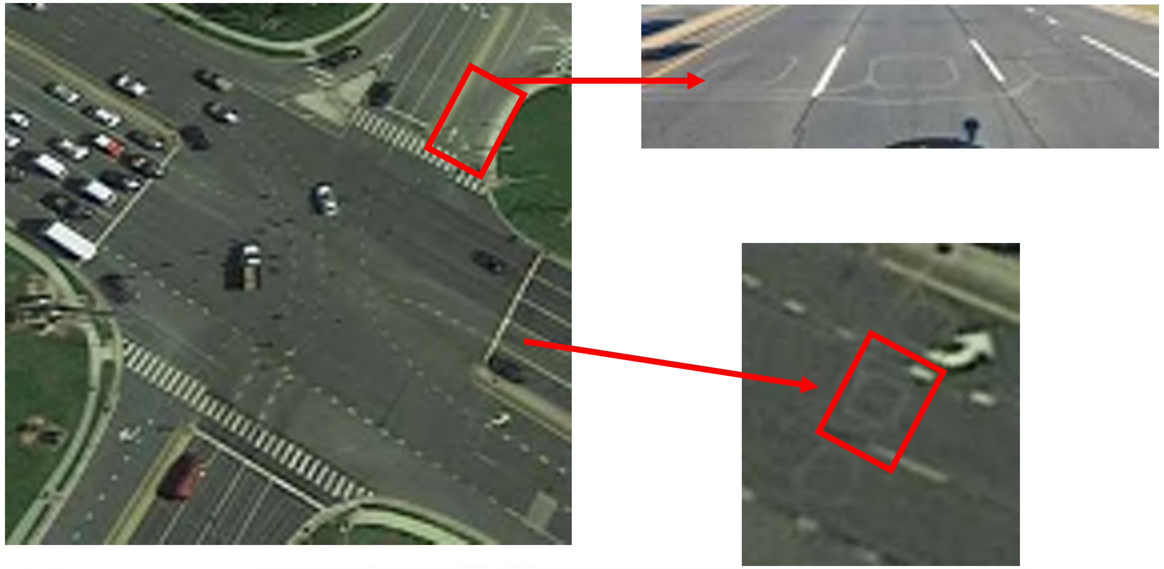
Signal Control Operations

- Signal patterns
  - Pre-coded and stored signal patterns
  - Common basic settings
  - Signal patterns vary by cycle length, phase split and offset
- Scheduled signal operation
  - Signal pattern changes based on time of day and day of week
  - Timetable based on historical traffic data
  - Serves recurring peak and off-peak traffic flows
  - Regions with typical and steady traffic patterns
- Dynamic/Responsive signal operation
  - Signal pattern change based on real-time traffic conditions
  - Serves non-recurring and fluctuating traffic pattern
  - Dynamic change rules and criteria
    - Respond to measured current conditions
- Respond to predicted future conditions

Proposed TOD Plan			
Hour	Min	Plan	Cycle
Everyday			
0	5	1/1/1	90 BAL
6	30	1/2/1	120 BAL
Weekdays (Days 2 - 6)			
7	30	2/1/1	150 SB
9	0	2/2/1	150 BAL
15	0	2/3/1	150 NB
20	0	1/2/1	120 BAL
Sunday (Day 1)			
9	30	2/2/1	150 BAL
13	30	3/3/1	165 NB
18	0	2/3/1	150 NB
22	30	1/2/1	120 BAL
Saturday (Day 7)			
7	30	2/2/1	150 BAL
11	0	3/1/1	165 SB
18	0	2/2/1	150 BAL
22	30	1/2/1	120 BAL

Siemens/Yunex Tactics - Dynamic Signal Operations

- Signal patterns change by responding to the traffic conditions measured by detector occupancy rate (o) and volumes (v)
- ***(V+O)%*** per direction per interval
  - Calculation based on per lane volumes and occupancy rate
  - Occupancy rate: percentage of time a vehicle is sensed over the detector
- ***v*** value per interval
  - Measure directional traffic flows
  - Identify congestion direction
  - Determine splits and offsets
- Trigger intersections
  - Seven intersections along the corridor
  - Detectors on downstream lanes



$$(V+O)\% = \frac{1}{n} \times \left(\sum_{d_i \in D} \frac{5\text{-min volume}_{d_i}}{150} \times 100 + \sum_{d_i \in D} 5\text{-min occupancy}_{d_i} \right) \%$$

where, d_i = perlane system detector i

$$D = \{d_1, d_2, d_3, \ldots, d_n\}$$

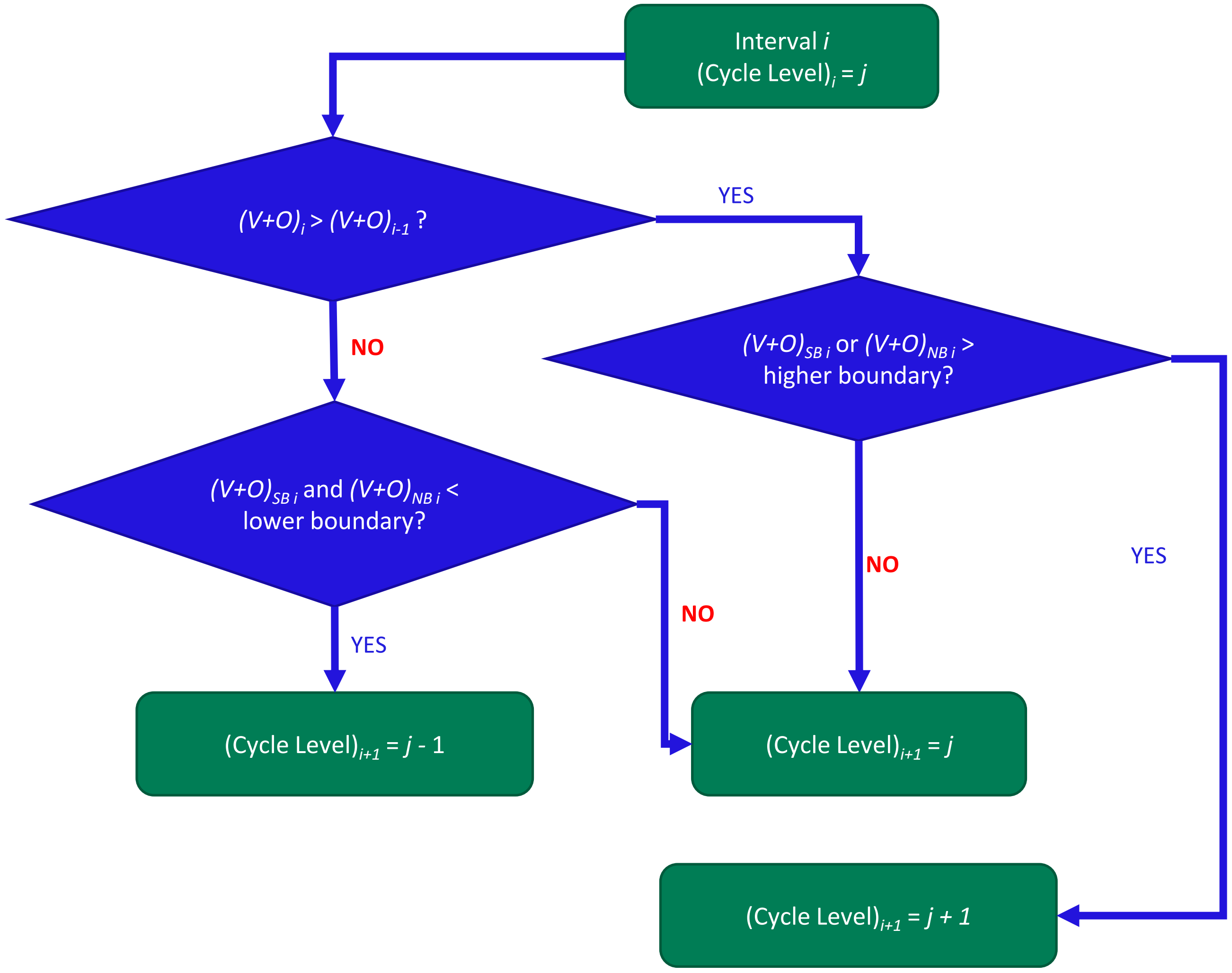
$$v\% = \frac{\frac{1}{m} \times \sum_{d_i \in DS} (SB\ volume)_{d_i}}{\left(\frac{1}{m} \times \sum_{d_i \in DS} (SB\ volume)_{d_i} + \frac{1}{n} \times \sum_{d_j \in DN} (NB\ volume)_{d_j}\right)}$$

where, d_i = perlane system detector i

$$DS = \{d_1, d_2, d_3, \ldots, d_m\}$$

$$DN = \{d_1, d_2, d_3, \ldots, d_n\}$$

Cycle Length and Split/Offset



Cycle Length – *(V+O)%*

- To change to longer cycle length
  - *V+O* is increasing
  - Either directional *V+O* greater than entry line
- To change to shorter cycle length
  - *V+O* is decreasing
  - Both directional *V+O* less than exit line

Offset – *v value*

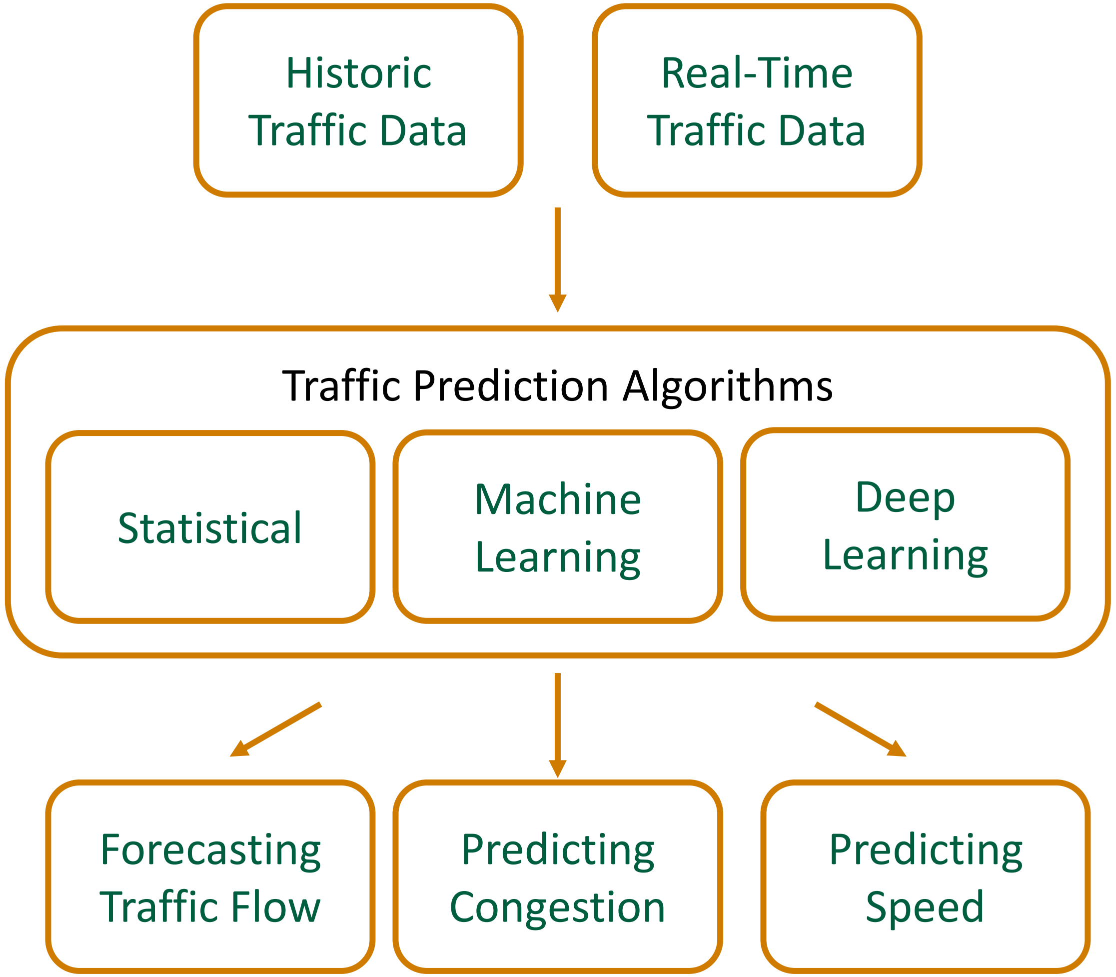
- Three levels of directionality
  - Default level = 1: balanced volumes at both directions
  - Level = 0: favor Northbound traffic
  - Level = 2: favor Southbound traffic

Example of Dynamic Signal Operation

Traffic Volume & Occupancy Prediction

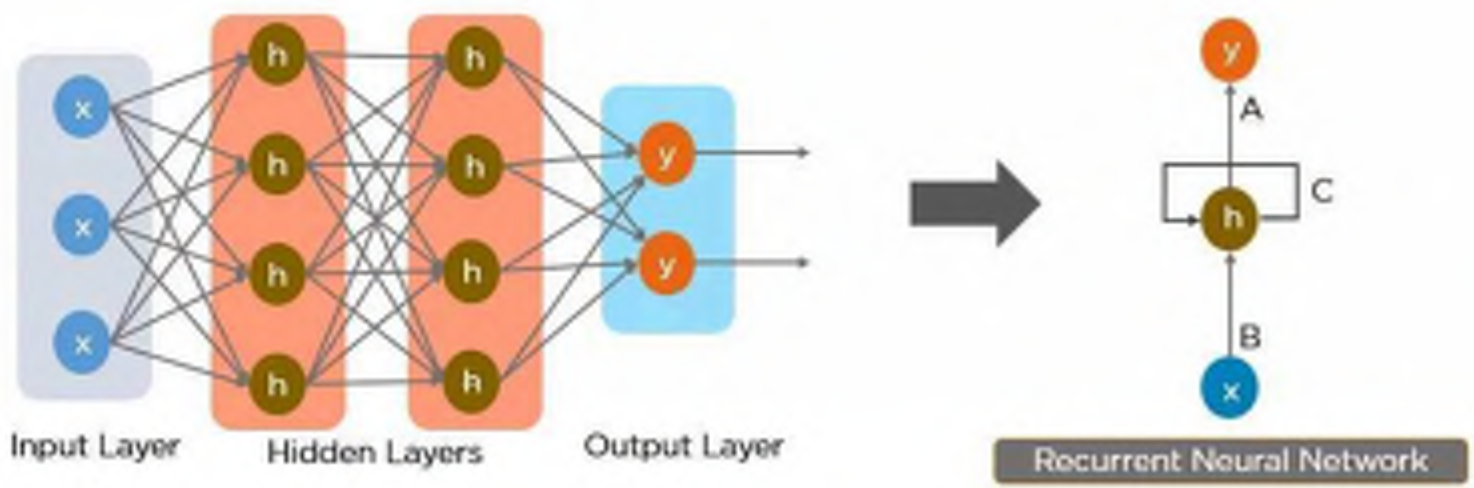
Traffic Data Prediction

- Traffic prediction
  - Forecasting the volume and density of traffic flow
- Traffic prediction algorithms
  - Statistical
    - Fast and cheaper but less accurate
    - Auto-Regressive Integrated Moving Average (ARIMA) model
  - Machine learning
    - Large masses of heterogeneous data
    - Random forest; k-nearest neighbors (KNN)
  - Deep Learning
    - Highly effective
    - Convolutional neural networks (CNNs); Recurrent neural networks (RNNs) – time series data



Deep Learning Model

- Neural network models
  - RNN (Recurrent Neural Networks)
    - Time-sequence data and prediction
    - With “memory” which remembers all information about what has been calculated.
    - Output of RNN depend on the prior elements within the sequence.
    - Types:
      - 1 – 1; 1 – many; many – 1; many - many
    - Vanishing gradient issue
  - LSTM
    - Long Short-Term Memory Networks
    - Address vanishing gradient issue



From: https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

Deep Learning Model – Data Selection

- Traffic Volume & Occupancy
  - Select key intersections in study area
  - Southbound and Northbound volume, southbound and northbound occupancy
  - Average volume & occupancy for all selected intersection by directions
- Summer weekends
- Data
  - Train dataset: Fridays from May to September 2022, except 7/15/2022 (13 days)
  - Test dataset: 7/15/2022
  - 24 hours by 5-min data point

X Inputs

Time	Average Volume
0:00:00	6.5
0:05:00	4.1
0:10:00	4.4
0:15:00	4.4
0:20:00	5.9
0:25:00	6.9
0:30:00	7.0
0:35:00	4.3
0:40:00	4.4
0:45:00	2.9
0:50:00	1.4
0:55:00	1.3

→ Recurrent Neural Network + LSTM Model →

Y Output

Time	Predicted Volume
1:00:00	1.4

Deep Learning Model – Data Clean Up & Processing

1. Data clean up & processing

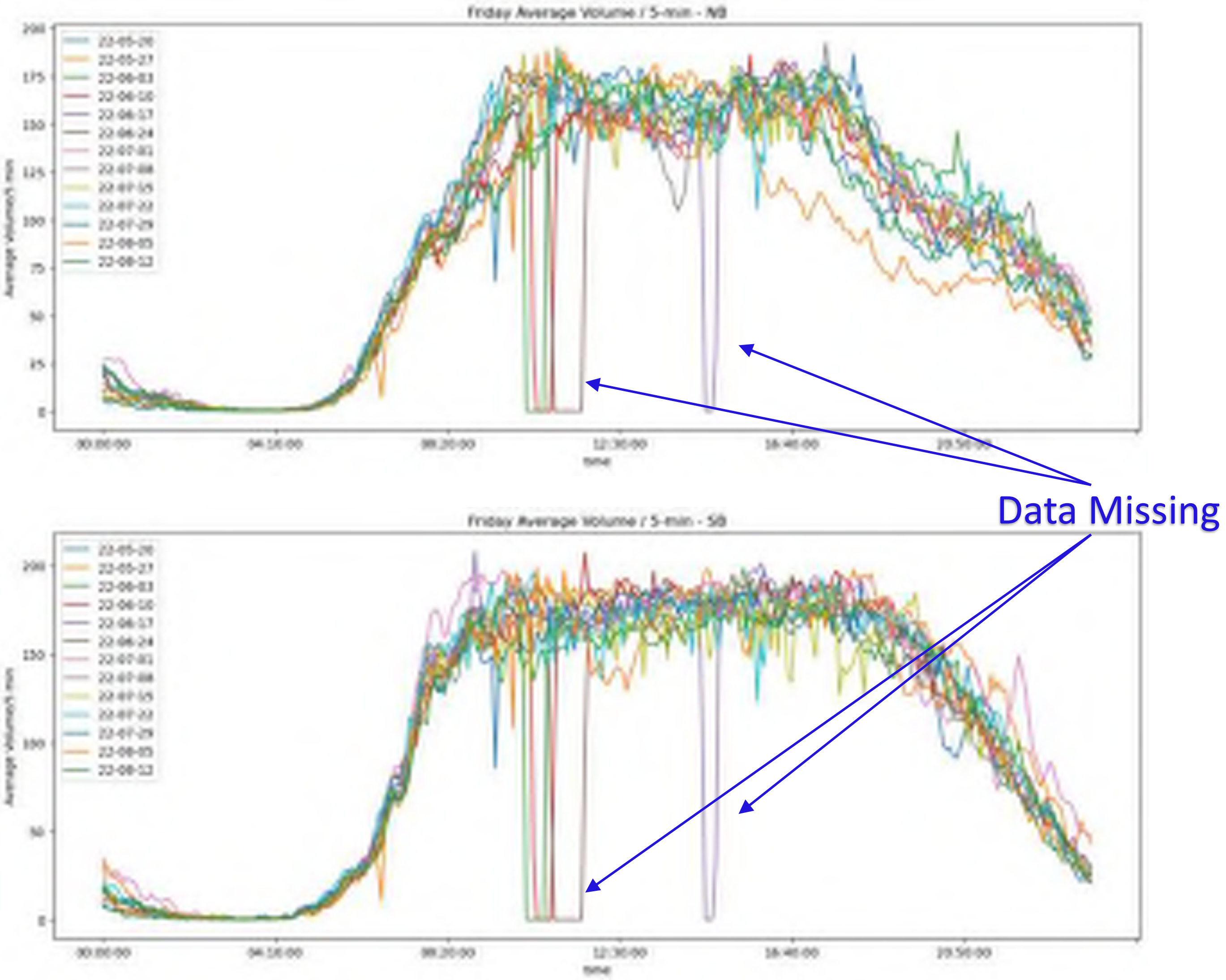
   1) ***Select*** Friday data
   2) ***Clean up and process*** missing & error data
   3) ***Split*** training set and testing set
   4) ***Calculate*** average data by selected intersections and directions for training set

      – Cycle pattern change

        - SB: 7 intersections
        - NB: 5 intersections

      – Offset pattern change

        - SB: 5 intersections
        - NB: 5 intersections



Deep Learning Model – Model Training & Testing

2. Deep Learning Model Training

- 4 LSTM layers with output layer
- “ADAM” optimizer
- Loss function: mean squared error
- TensorFlow – keras / scikit-learn - MinMaxScaler
- 6 deep learning models
  - Cycle change – volume : SB + NB
  - Cycle change – occupancy : SB + NB
  - Offset change – volume: SB + NB

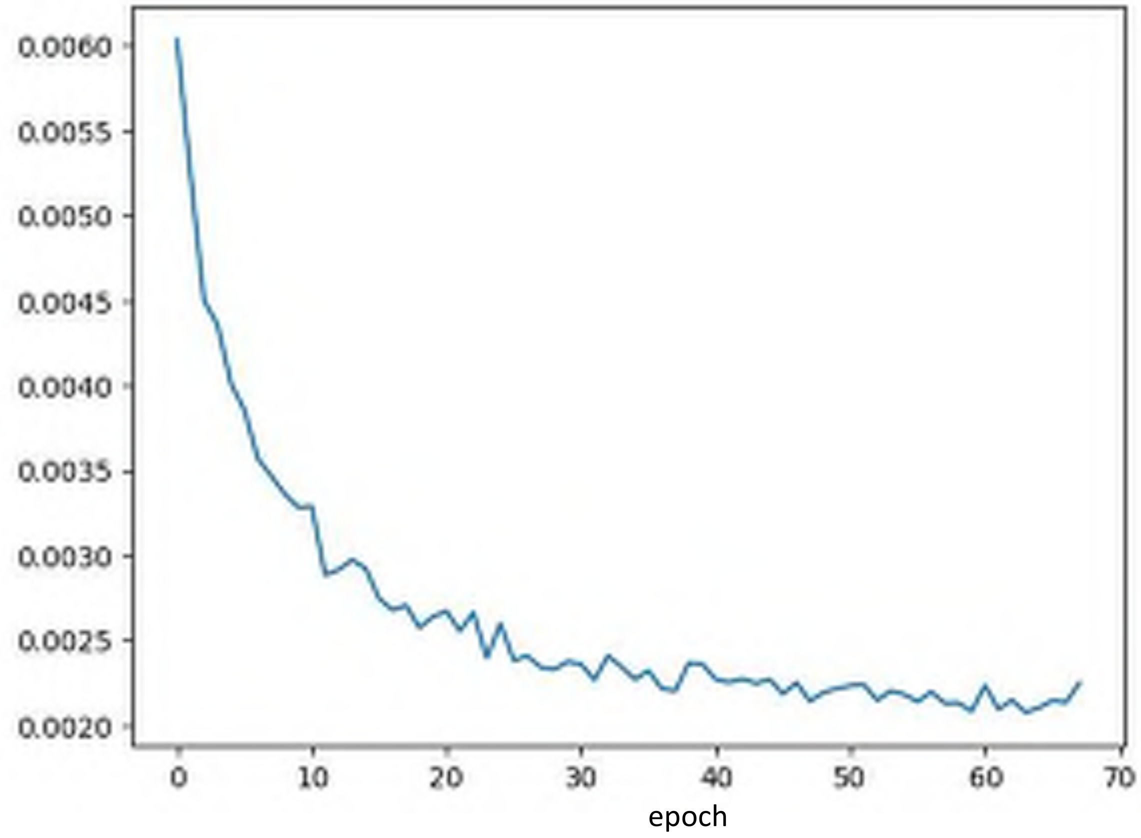
3. Deep Learning Model Testing

- Use trained model to test 7/15/2022 data
- Run and test multiple attributes with multiple epochs

4. Deep Learning Model Save

- Pickle format

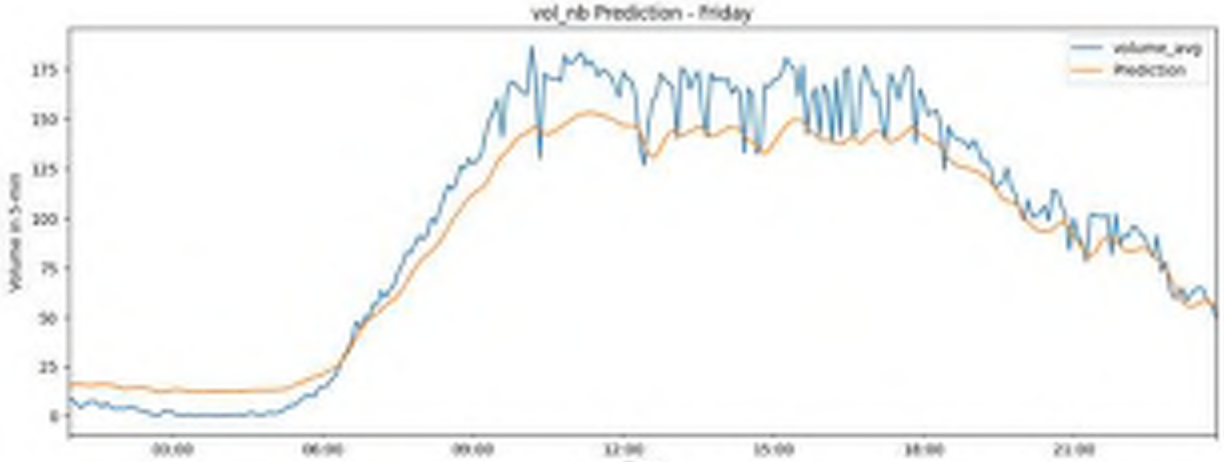
Training Model Loss – Northbound Volume (Cycle Change)

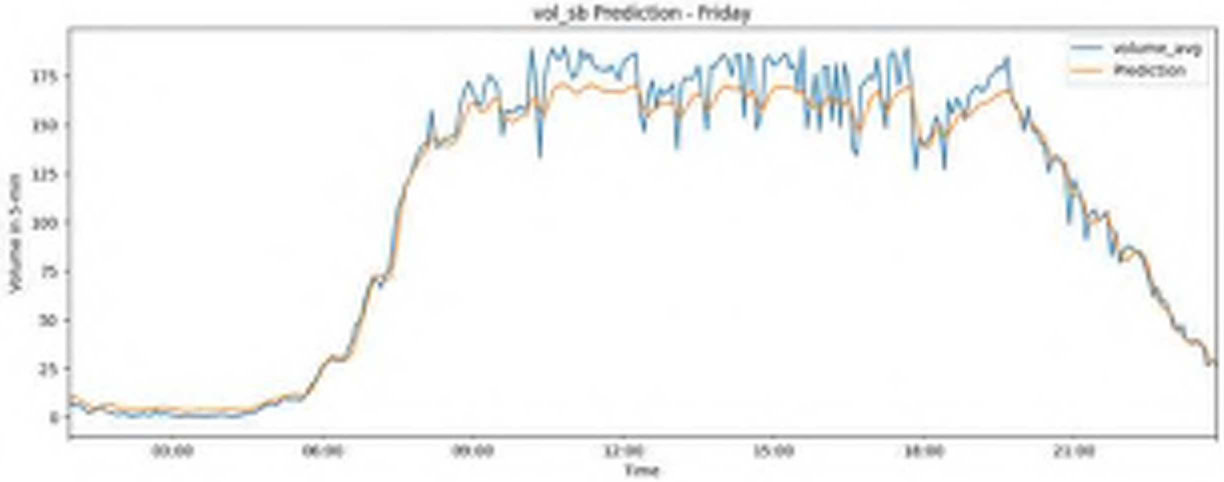


Deep Learning Model – Prediction

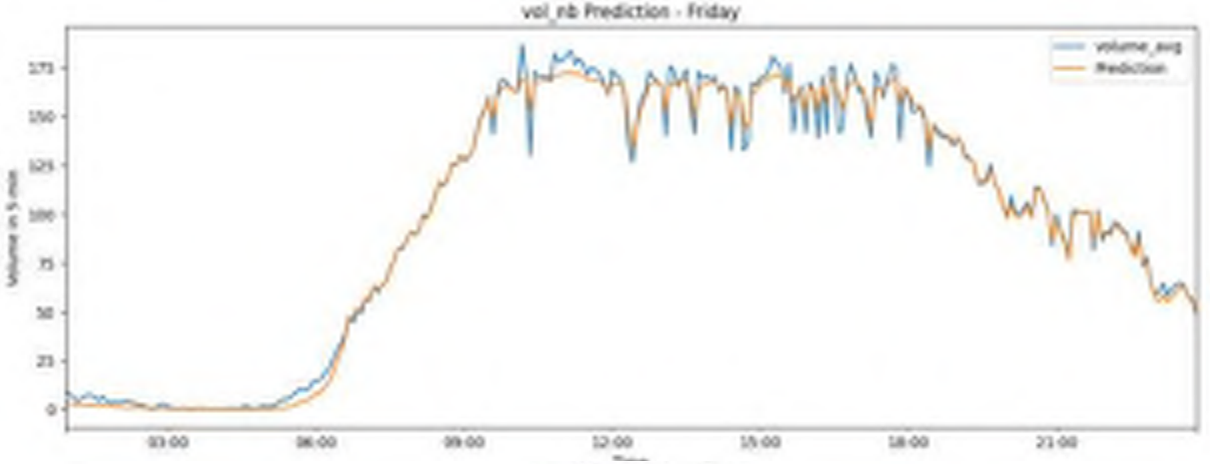
- Traffic Volume & Occupancy

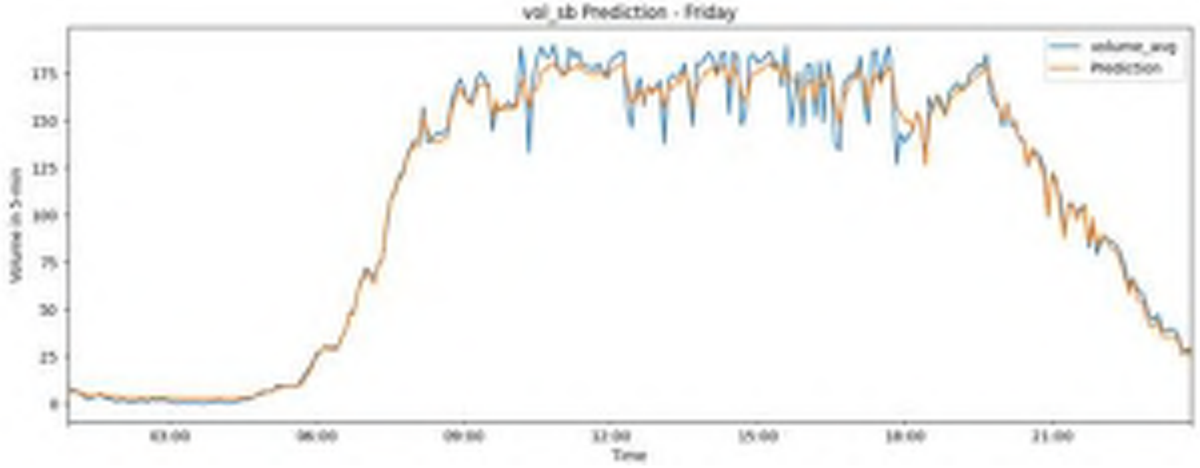
Before





After

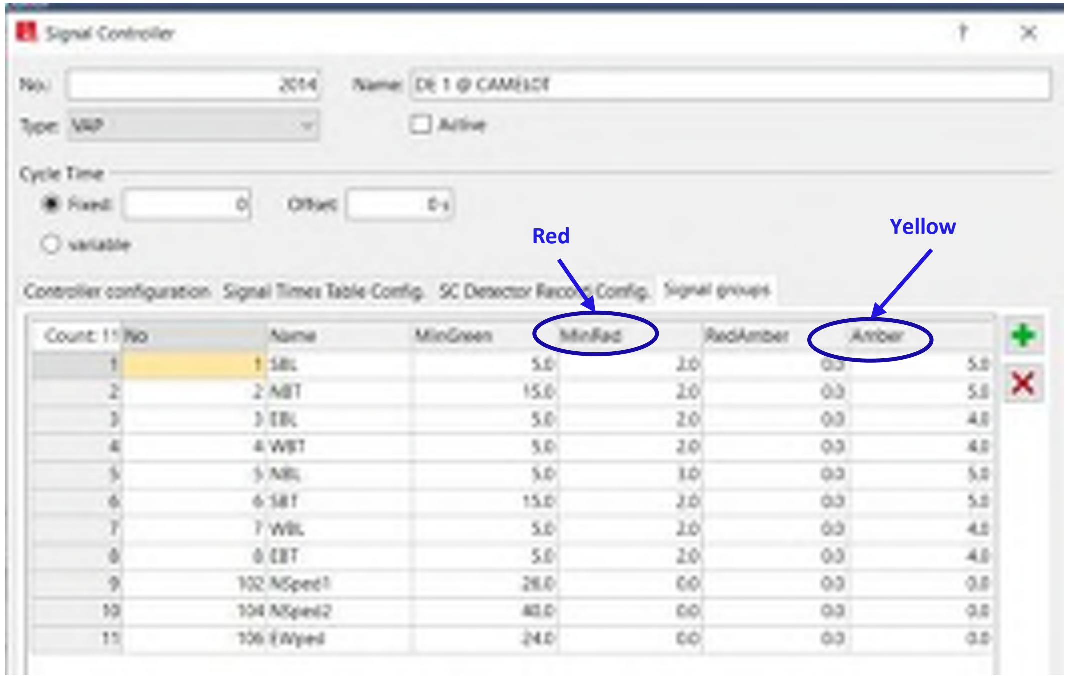


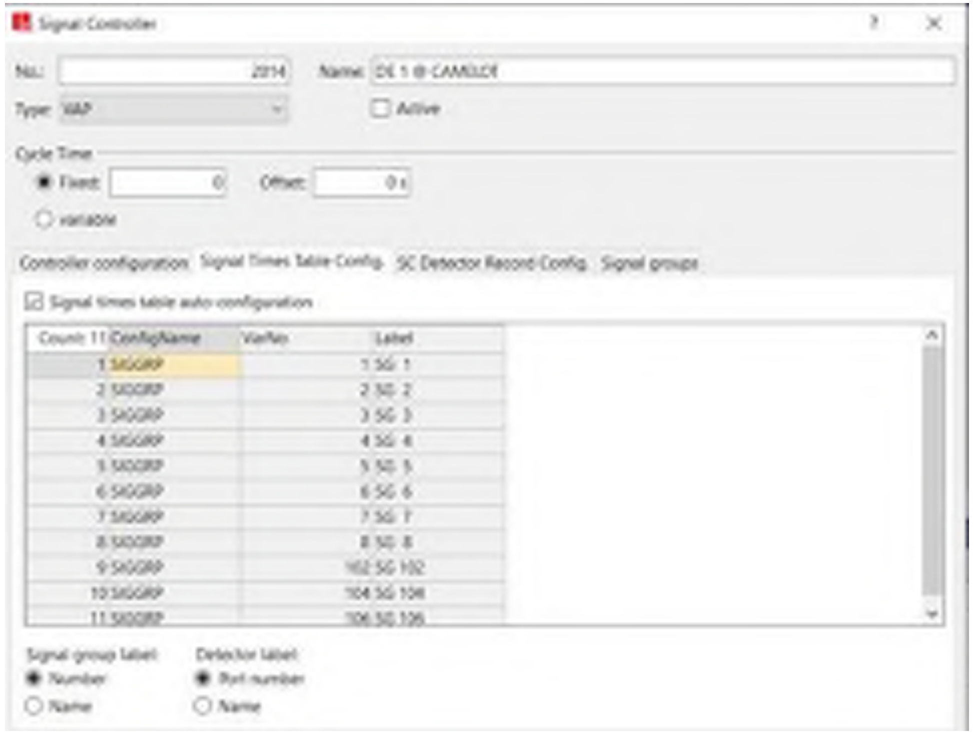


Signal Controller and API Development

VAP Controller

- VAP Controller: traffic dependent programming
  - Simulate programmable vehicle-actuated signal controls
  - Interpret control logic commands and generate signal control commands
  - Stage- or group-based





- In VISSIM
  - Pre-coded basic signal group settings for all patterns
  - Signal patterns saved in text files
    - **.pua*: signal data file
    - **.vap*: program logic

Files of VAP Controller

- PUA file:
  - SIGNAL_GROUPS: phase groups
  - STAGES: signal operational statuses
  - STARTING_STAGE: first operational status
- VAP file
  - Signal group concept
  - Define signal patterns and recall modes
  - All patterns have the same phases sequence

Phase split

```
ARRAY
    MAXGREEN[9,8] = [[25,84,0,5,25,84,0,5],[25,114,0,5,25,114,0,5],[25,144,0,5,25,144,0,5],
                     [25,144,0,5,25,144,0,5],[25,144,0,5,25,144,0,5],[25,144,0,5,25,144,0,5],
                     [25,159,0,5,25,159,0,5],[25,159,0,5,25,159,0,5],[25,159,0,5,25,159,0,5]],
    /* maxgreen is basically the max green time of each phase in each pattern */
    MINGAP10[8] = [30,50,0,40,30,50,0,40],  /* min gaps are entered in tenths of secs */
    MAXRECALL[8] = [0,0,0,0,0,0,0,0],
    MINRECALL[8] = [0,1,0,0,0,1,0,0],          /* MAXRECALL and MINRECALL cannot be true at the same time */
    PEDPHASE[8] = [0,1,0,1,0,1,0,0],           /* indicate if the phase has pedestrian phase */
    PEDRECALL[8] = [0,0,0,0,0,0,0,0],          /* indicate if the ped phase has ped-recall */
    OFFSET[9] = [62,60,26,5,137,0,26,5,137],                   /* This number is the (RBC offset - first_phase_split_time) */
    VEHDETECTOR[8] = [1,0,0,1,1,0,0,1],        /* if the movement has detectors. */
    CYCLE[9] = [90,120,150,150,150,150,165,165,165],              /* cycle length of patterns */
    YellowRed[8] = [6,6,0,6,6,6,0,6],          /* yellow time + red time of each phase */
    Conflicts[8,4] = [[2,4,8,8],[1,4,8,8],[0,0,0,0],[1,2,5,6],[4,6,8,8],[4,5,8,8],[0,0,0,0],[1,2,5,6]],
                                               /* define the conflict phases/movements of current phase/movement */
```

```
$SIGNAL_GROUPS
$
SBL             1
NBT             2
EBL             3
WBT             4
NBL             5
SBT             6
WBL             7
EBT             8
NSped1        102
EWped         104
NSped2        106


$STAGES
$
stage_1   SBL NBL
red       NBT EBL WBT SBT WBL EBT NSped1 EWped NSped2
stage_2   NBT NBL
red       SBL EBL WBT SBT WBL EBT NSped1 EWped NSped2
stage_3   NBT SBT
red       SBL EBL WBT NBL WBL EBT NSped1 EWped NSped2
stage_4   EBL WBL
red       SBL NBT WBT NBL SBT EBT NSped1 EWped NSped2
stage_5   WBT WBL
red       SBL NBT EBL NBL SBT EBT NSped1 EWped NSped2
stage_6   WBT EBT
red       SBL NBT EBL NBL SBT WBL NSped1 EWped NSped2

$STARTING_STAGE
$
stage_1

$END
```

VISSIM COM Application

- Developed in Python 3.9 script
- Identify signal controller and change its pattern number
- Use VISSIM event-based script
- Text files of inputs:
  - List of intersections and responsive zone
  - List of trigger intersections for collecting measurements
  - List of pre-coded signal patterns
  - *(V+O)%* threshold for selecting cycle length
  - *v* value threshold and changing matrix for selecting directionality
- Output: VISSIM MOEs

zone	V+O threshold	lower_threshold	high_threshold	exit_cycle	entry_cycle
S020	1	3	8	90	120
S020	2	10	18	120	150
S020	3	30	40	150	165

zone	cycle	direction	pattern_id	vap_prg_no
S020	90	BAL	111	1
S020	120	BAL	121	2
S020	150	SB	211	3
S020	150	BAL	221	4
S020	150	NB	231	5
S020	150	NB SLOW	232	6
S020	165	SB	311	7
S020	165	NB	331	8
S020	165	BAL	321	9

zone	int_id
S020	2005
S020	2007
S020	2010
S020	2014
S020	2016
S020	2017
S020	2026

```
def main():
    time = Vissim.Simulation.AttValue('SimSec')
    # 1. get v+o information from the data collection measurement
    if (time > EVAL_TIME_STEP) and (time % EVAL_TIME_STEP == 1):
        for zone in TARGET_ZONE:
            result = get_vo_vol_info(zone)

            # 2. check v+o calculation results with thresholds an
            #    check if there are any changes
            proposed_vap_prg, new_cycle, new_offset_level
                = cycle_offset_check(result, zone)
            # 3. get the v+o thresholds and
            #    get the zone-20 target pattern number
            try:
                if current_prg_no[zone] != proposed_vap_prg:
                    switch_vap_prg_zone(zone, proposed_vap_prg)
                    cur_offset_level[zone] = new_offset_level
                    cur_cycles[zone] = new_cycle
            except:
                print("Change VAP program error.")

            vo_vol_track(result, zone, time, proposed_vap_prg)
    return
```

Analysis Results

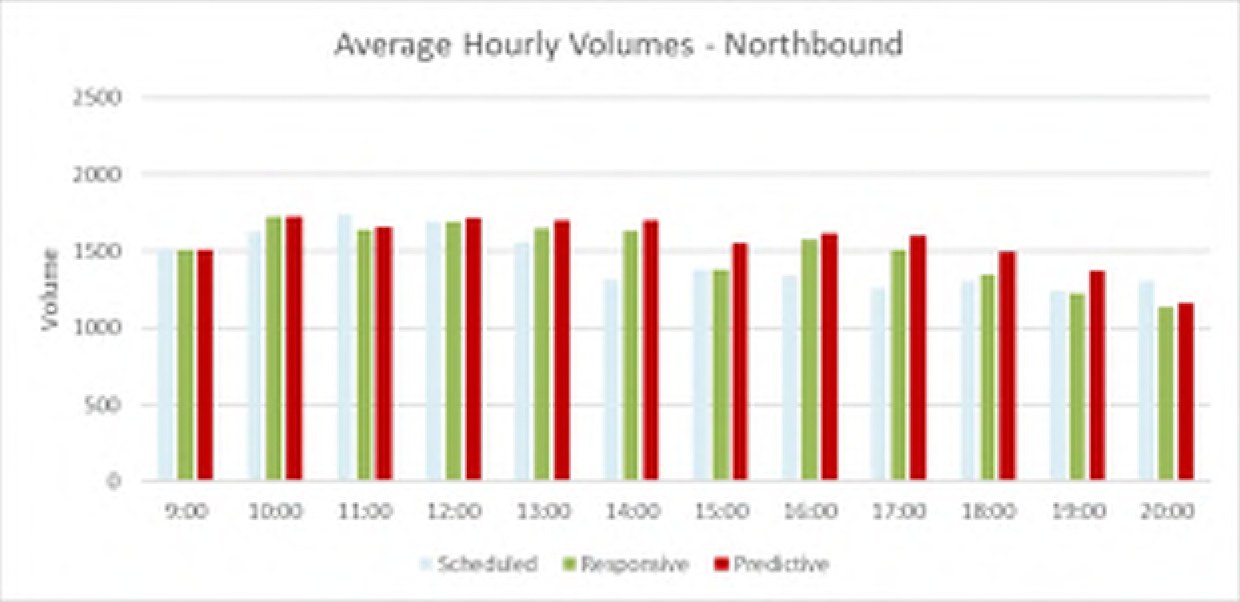
Simulation Assumption

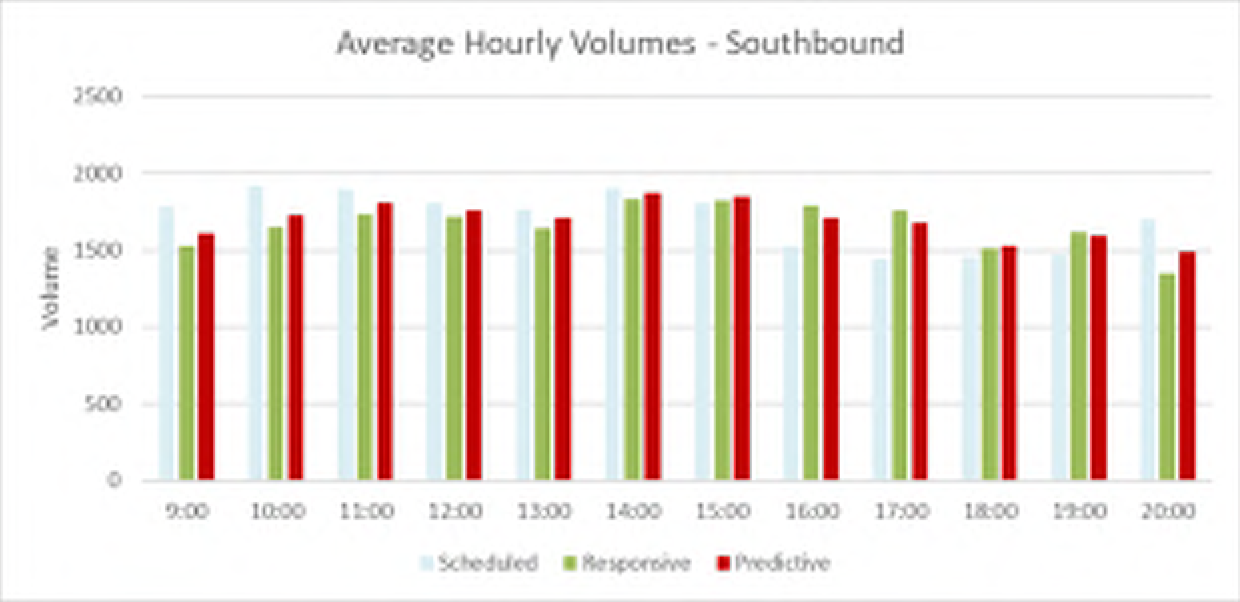
- Simulation period: 9:00 AM – 9:00 PM
- One hour seeding time
- Scenarios
  - **Scheduled**: signal pattern change follows preset timetable.
  - **Responsive**: signal pattern change corresponds *to (V+O)%* and *v%* value of current interval in simulation run.
  - **Predictive**: signal pattern change corresponds to (V+O) % and v% value of deep learning model predicted values based on simulation output.
- Vehicle volume input: 15-min interval
- Trigger measurements
  - Captured using Data Collection Measurements (DCM)
  - Volume and occupancy rate: 5-min interval
- Output MOEs
  - Arterial throughputs: 15-min interval, aggregated by one hour
  - Corridor travel time: 1-hour interval
  - Intersection operation: 1-hour interval

VISSIM Network

Simulation Results - *Friday*

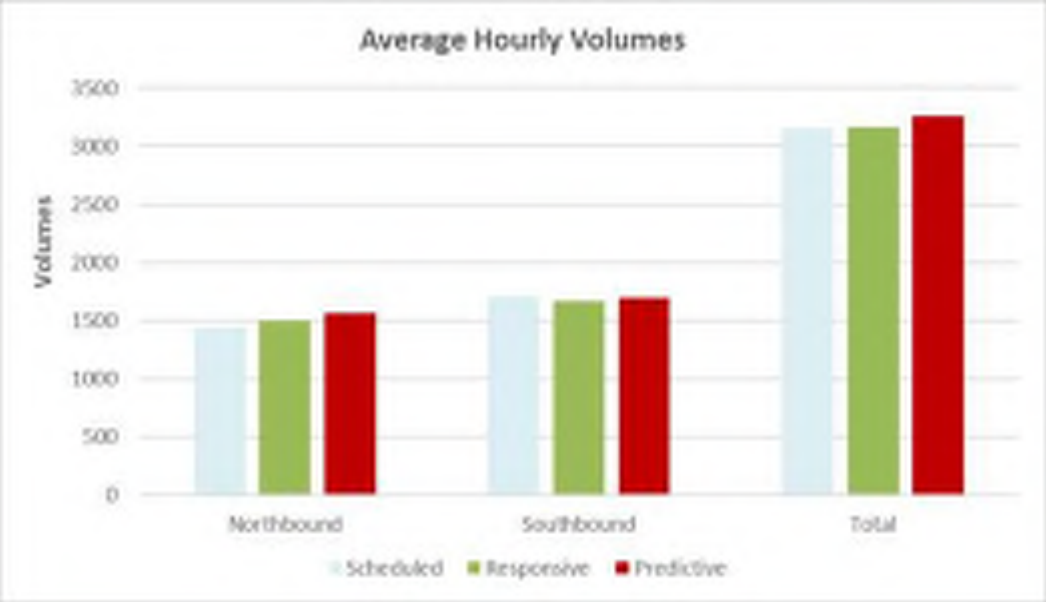
- Comparable corridor throughputs
- Dynamic pattern switch improved
  - Intersection operation
  - Corridor travel time

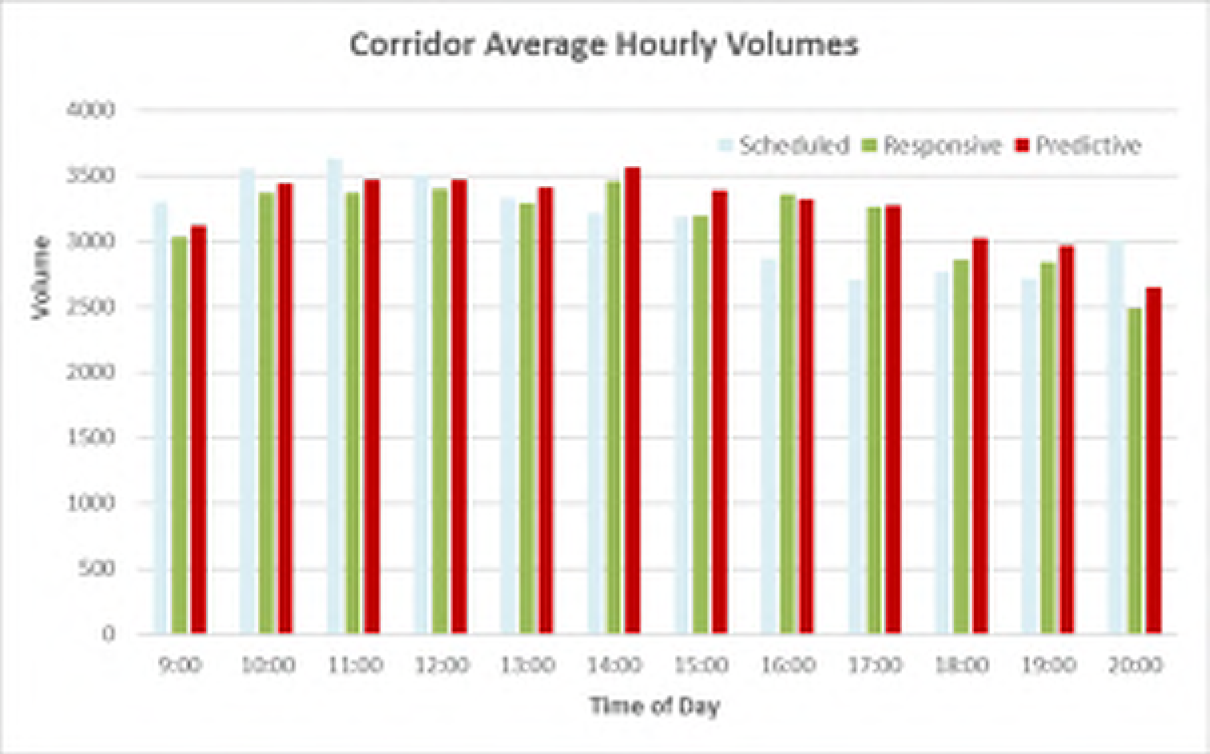


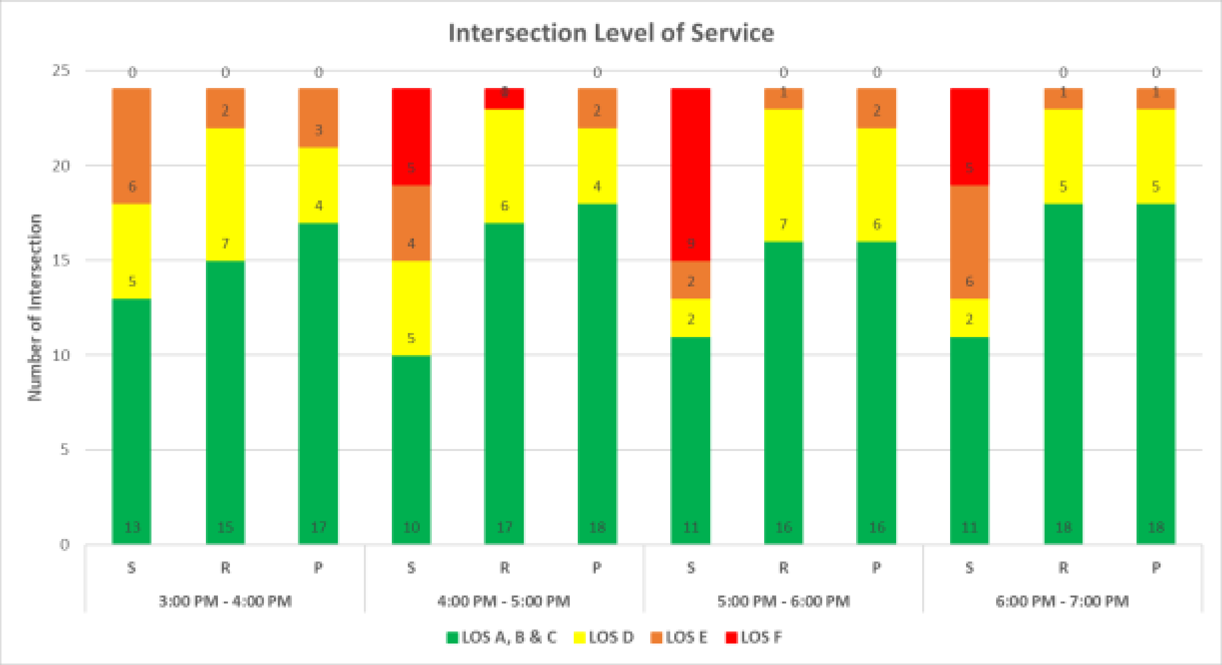


Simulation Results - *Friday*

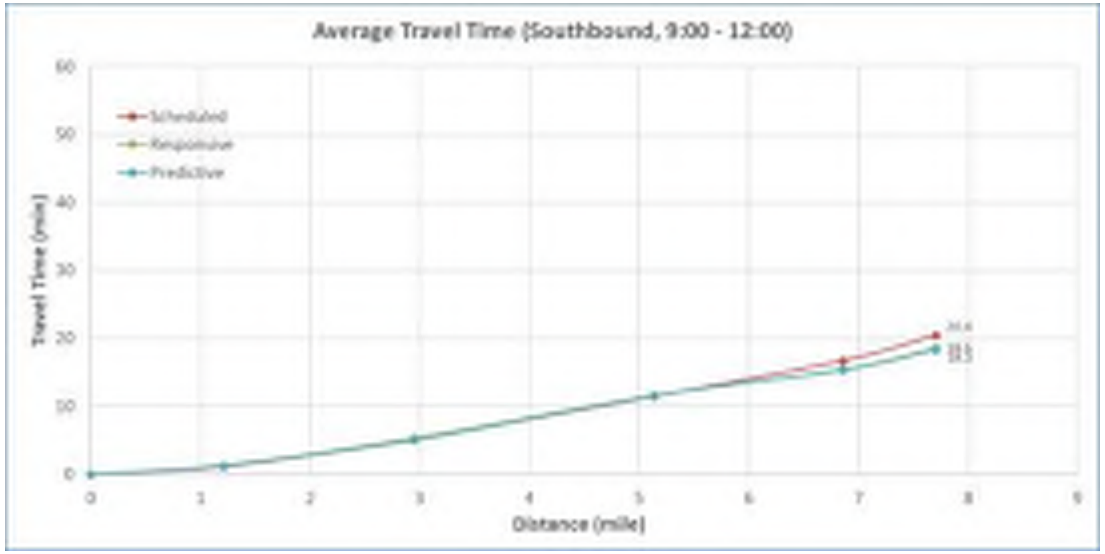
- Comparable corridor throughputs
- Dynamic pattern switch improved

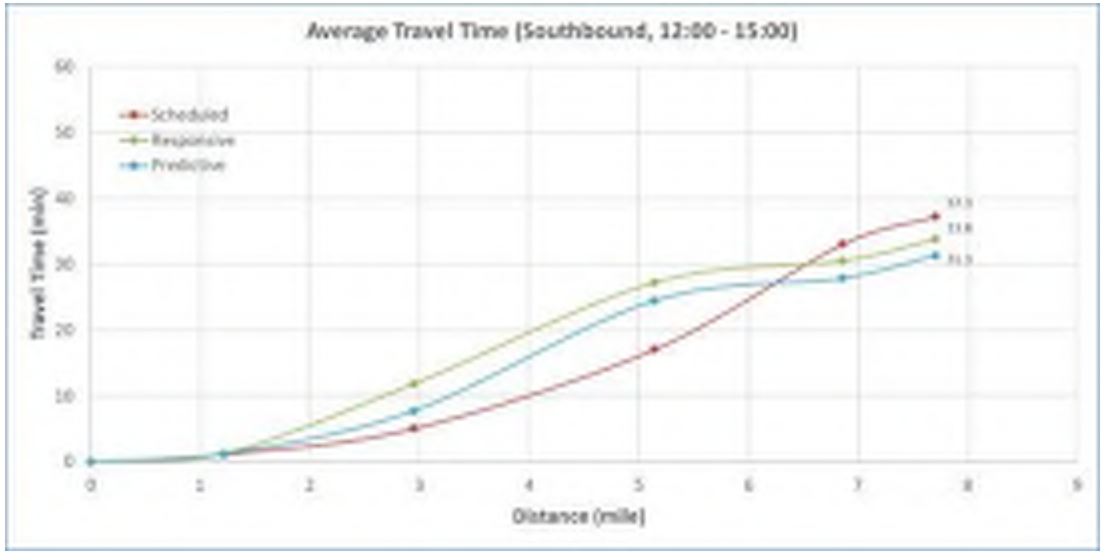


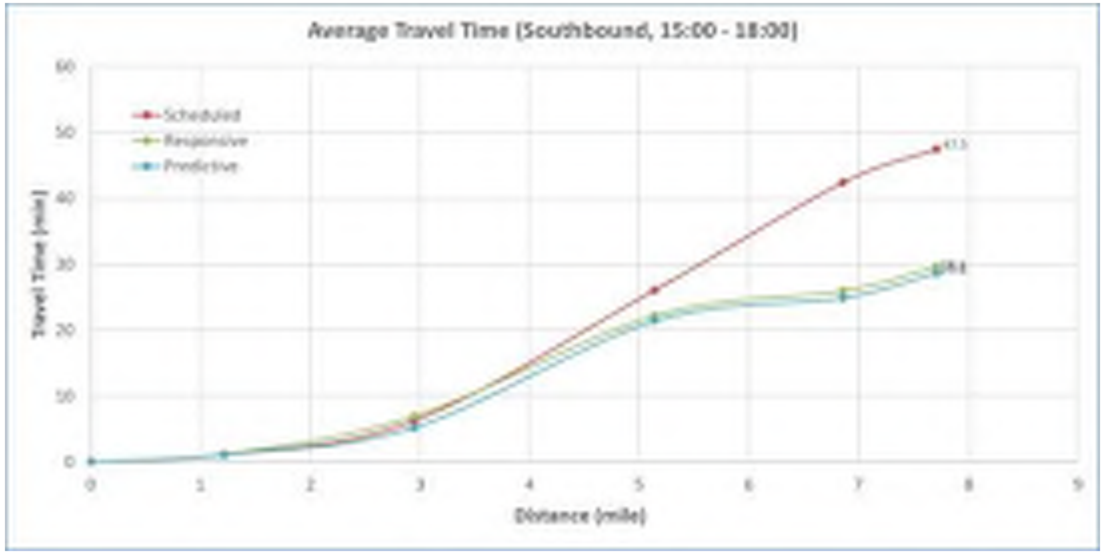


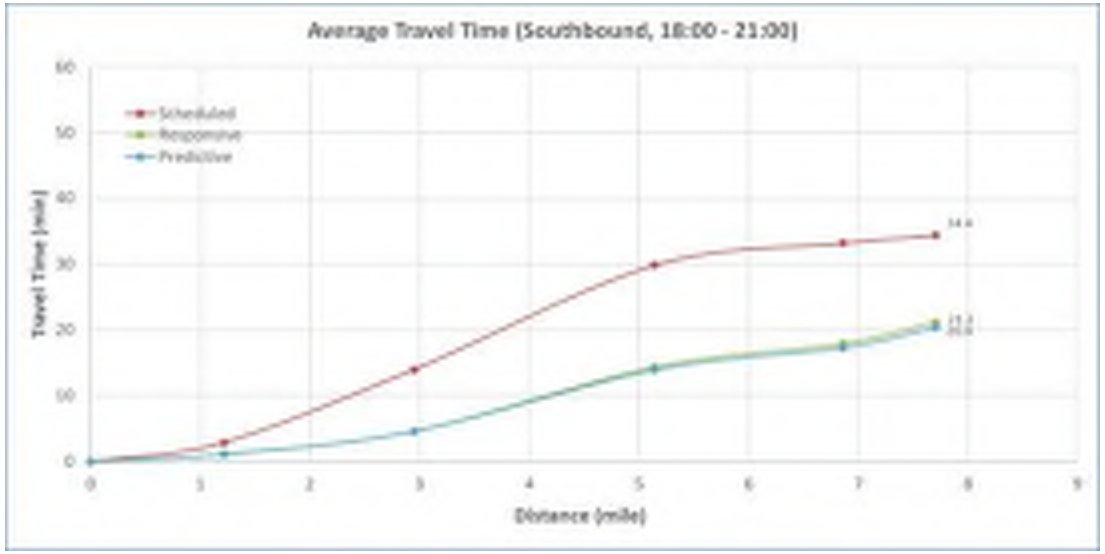


Travel Time Comparison - Friday - Southbound

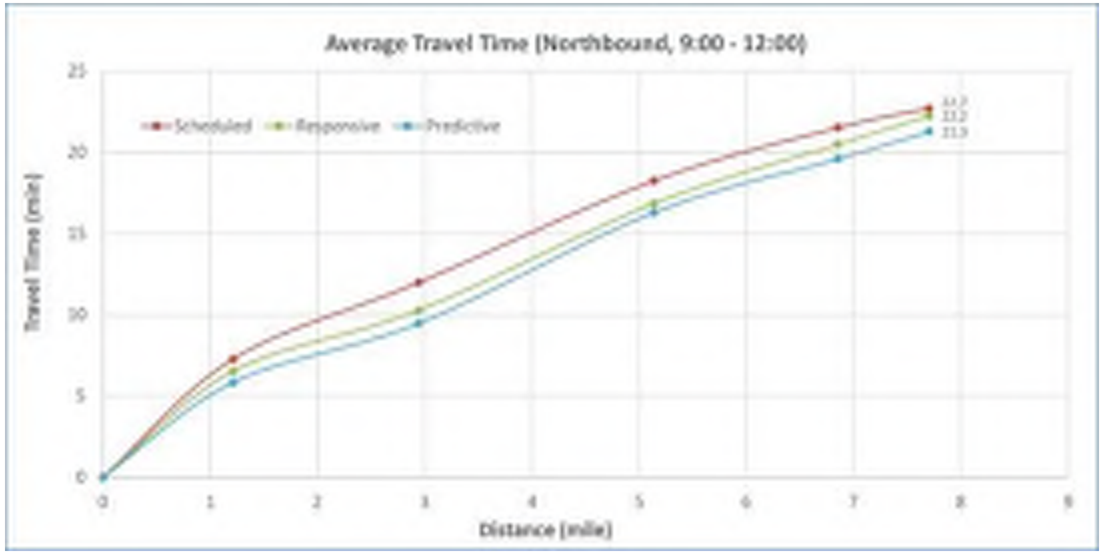


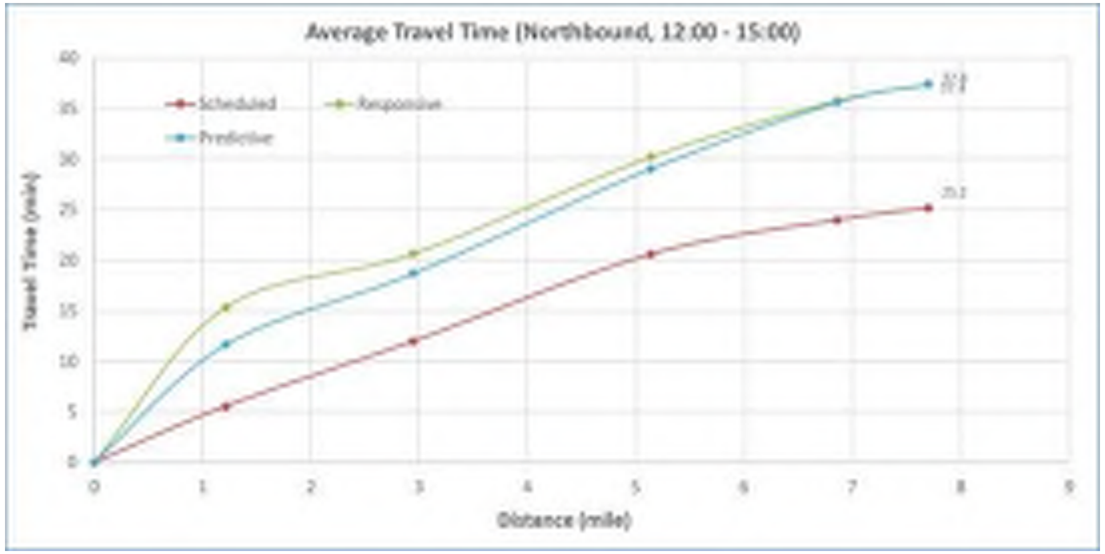


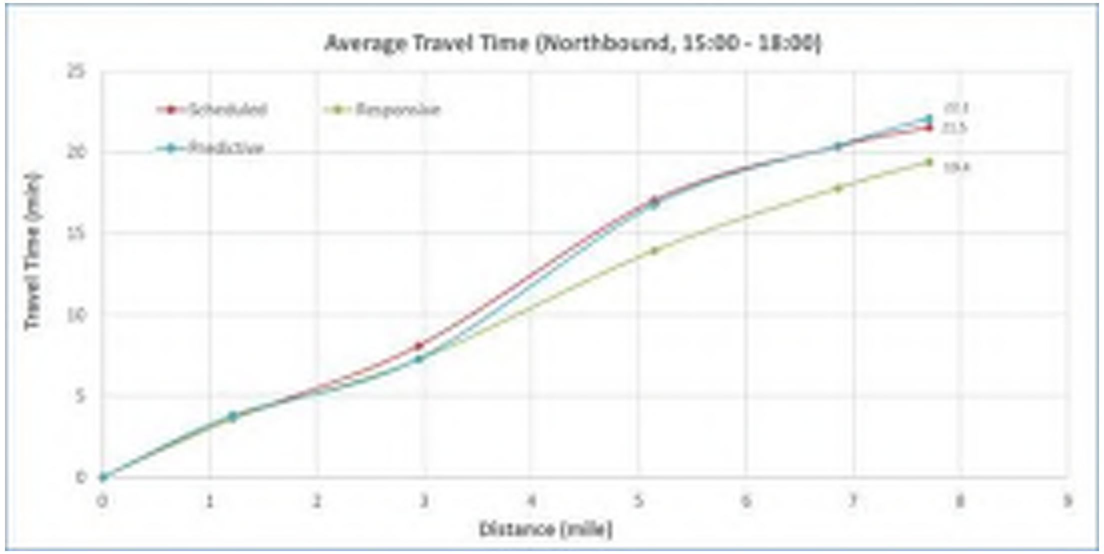


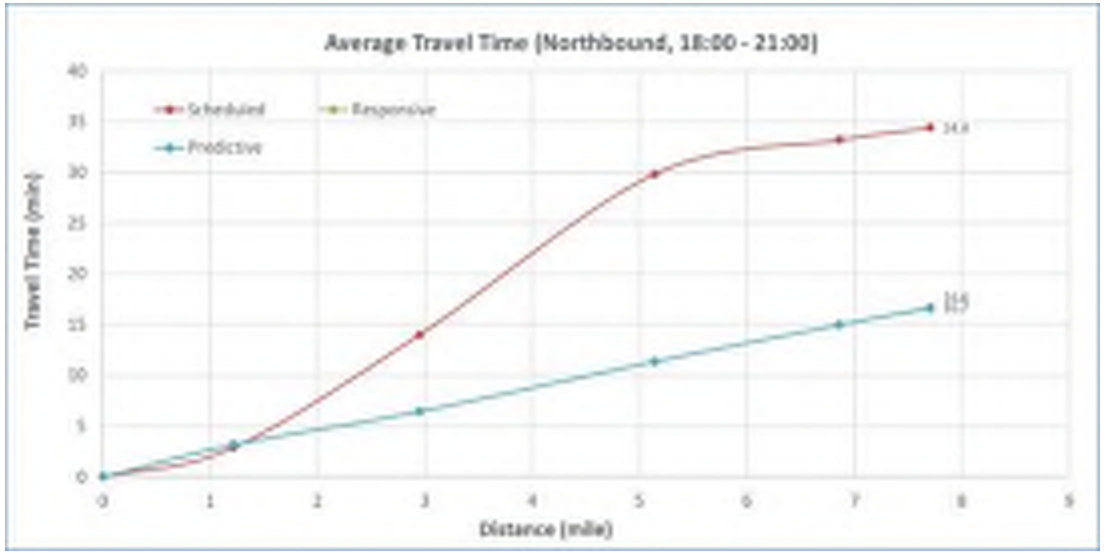


Travel Time Comparison - Friday - Northbound



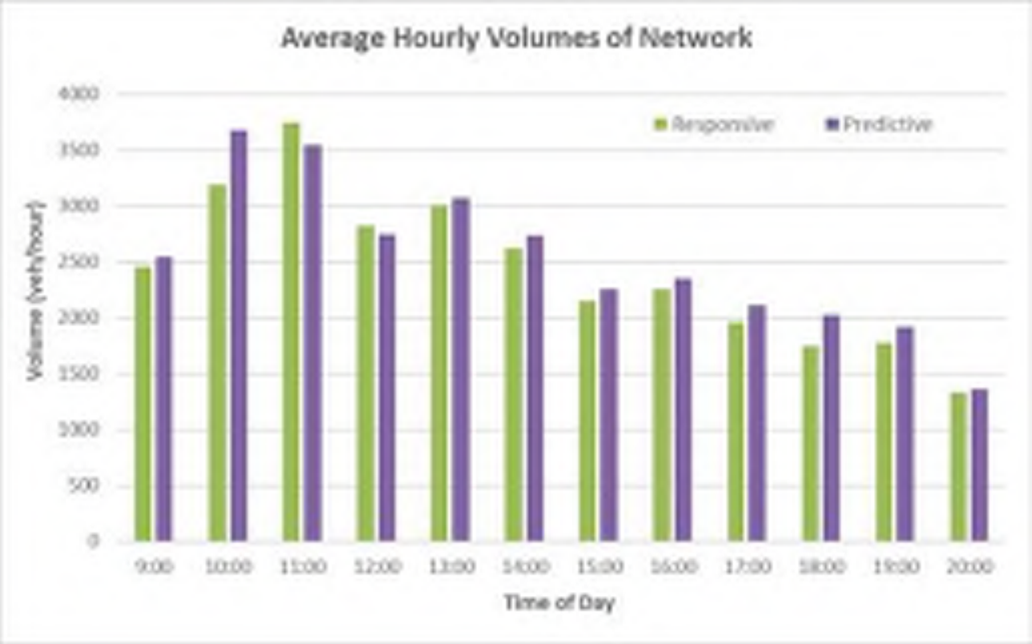


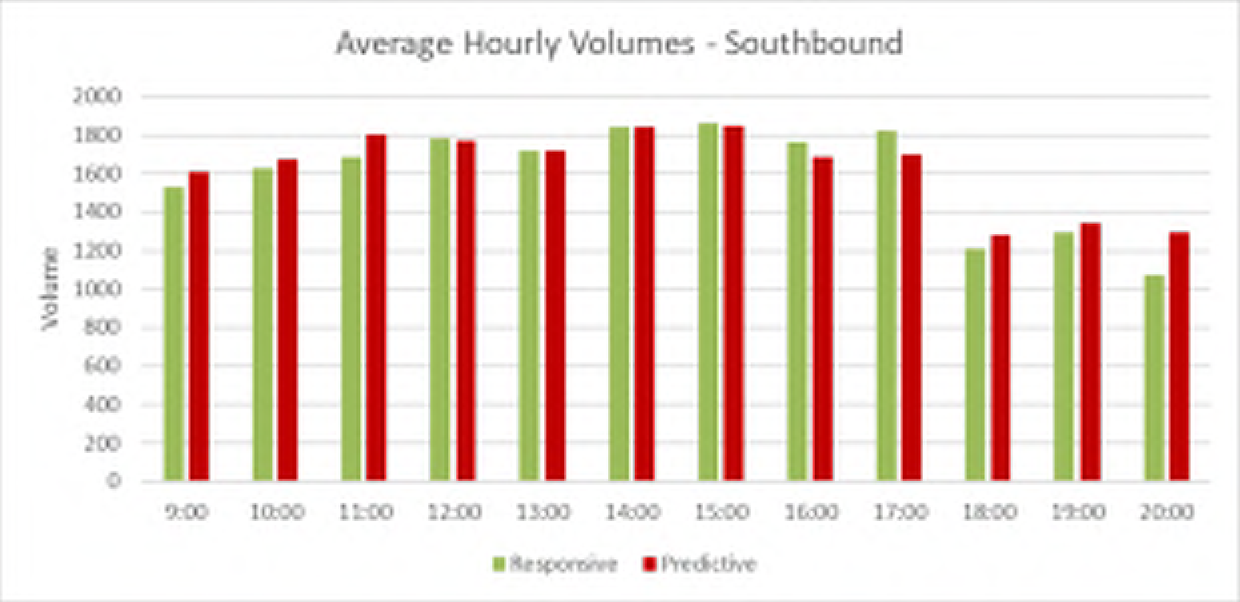


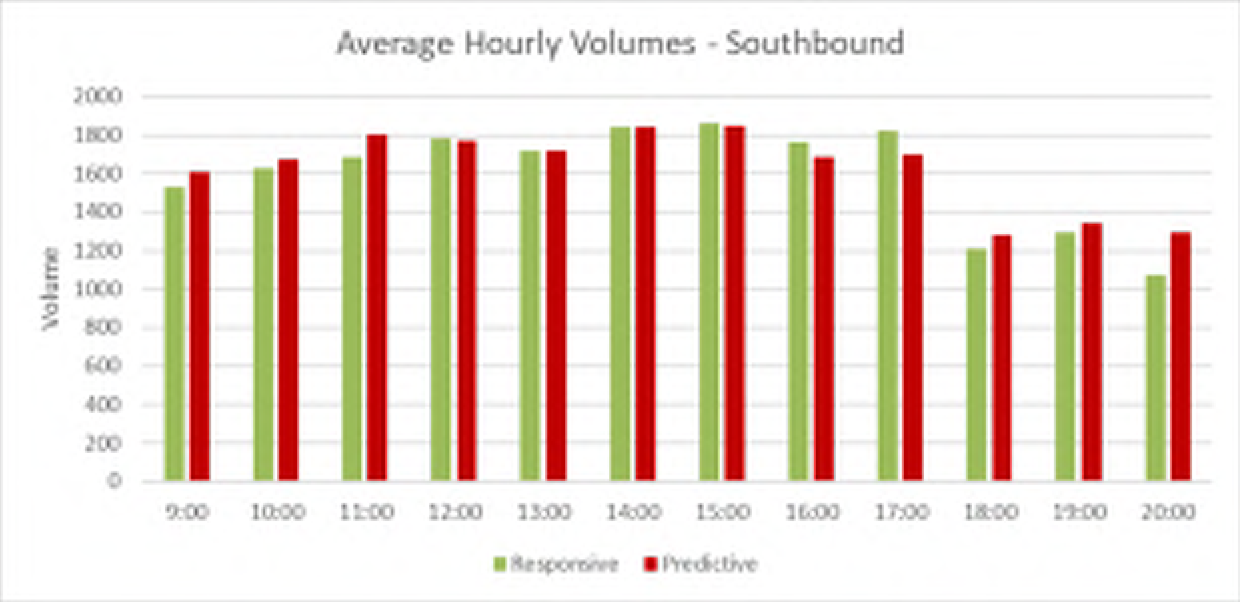


Simulation Results - *Saturday*

- Comparable corridor throughputs
- Dynamic pattern switch improved
  - Intersection operation
  - Corridor travel time

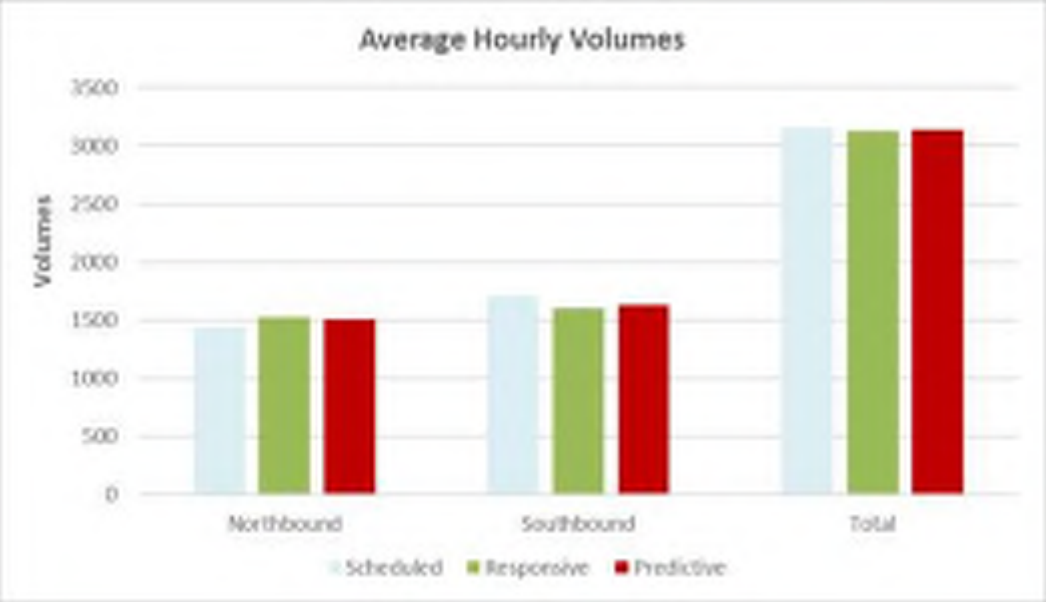


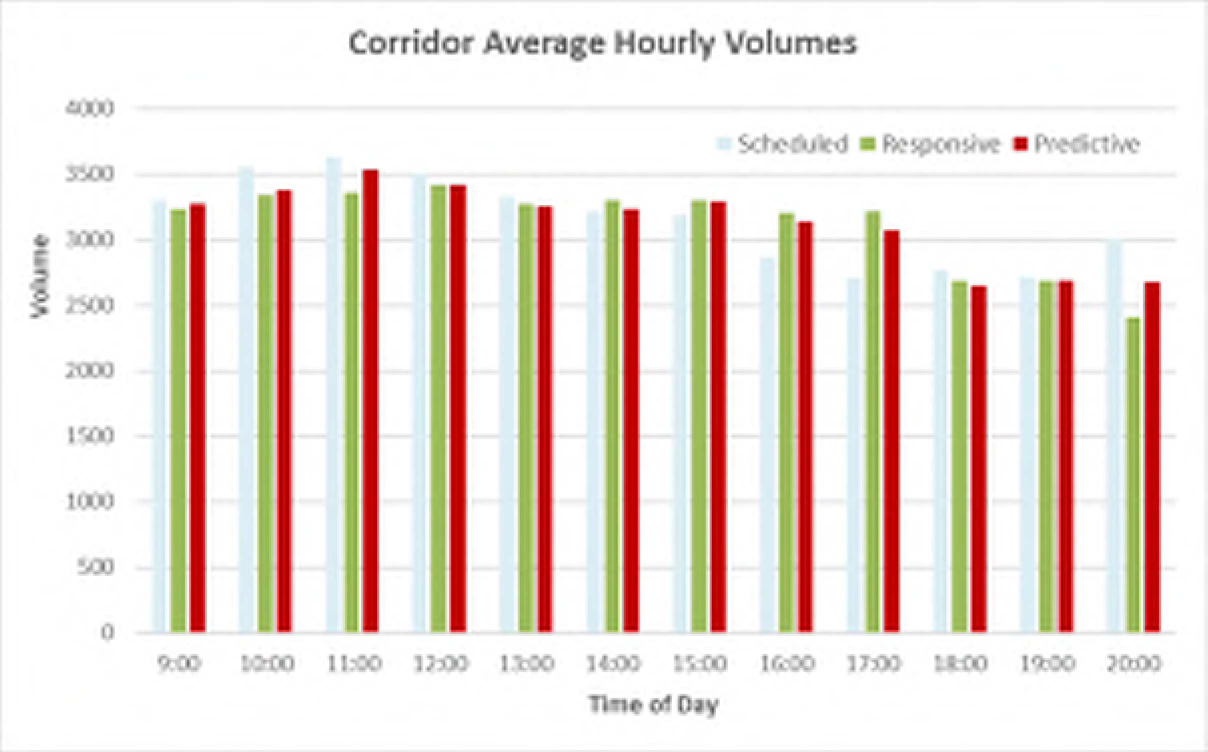


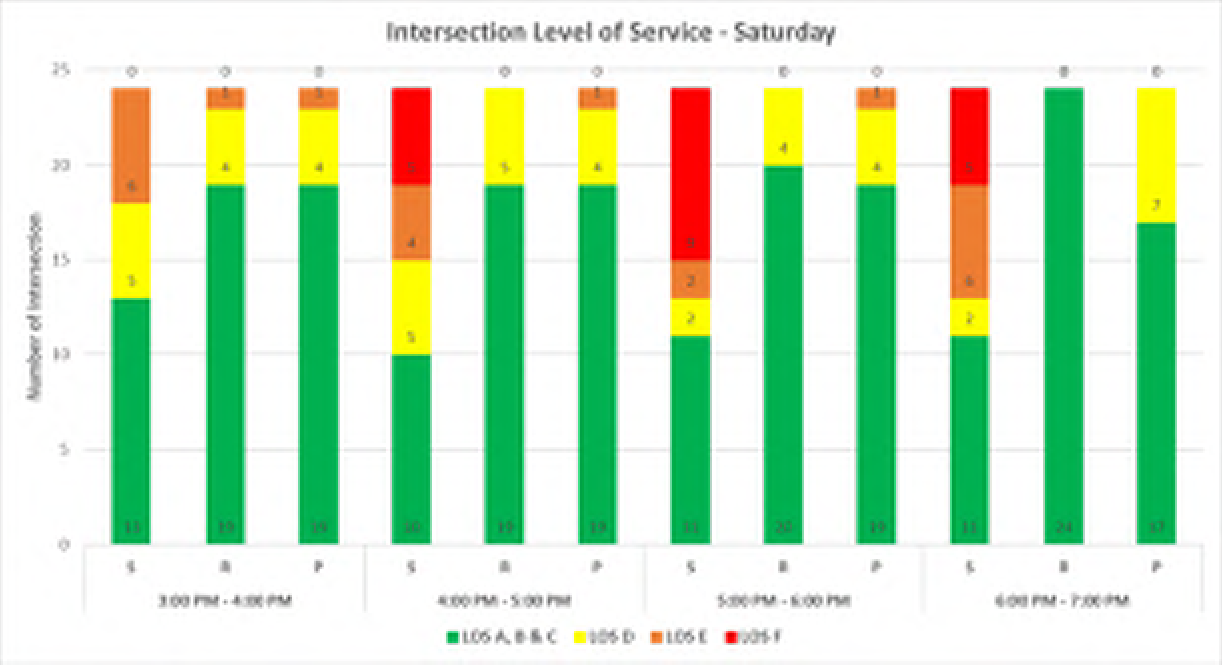


Simulation Results - *Saturday*

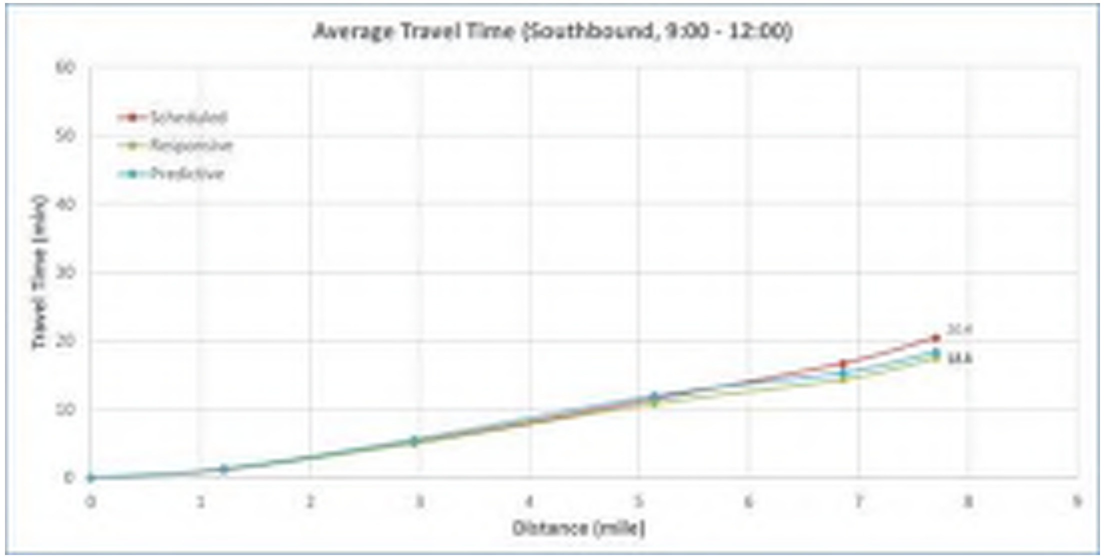
- Comparable corridor throughputs
- Dynamic pattern switch improved

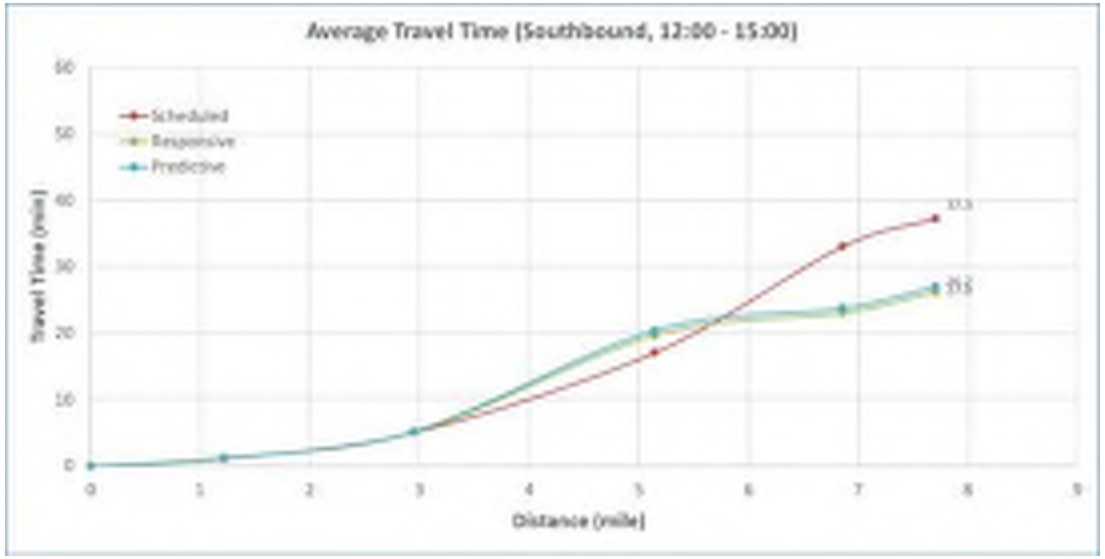


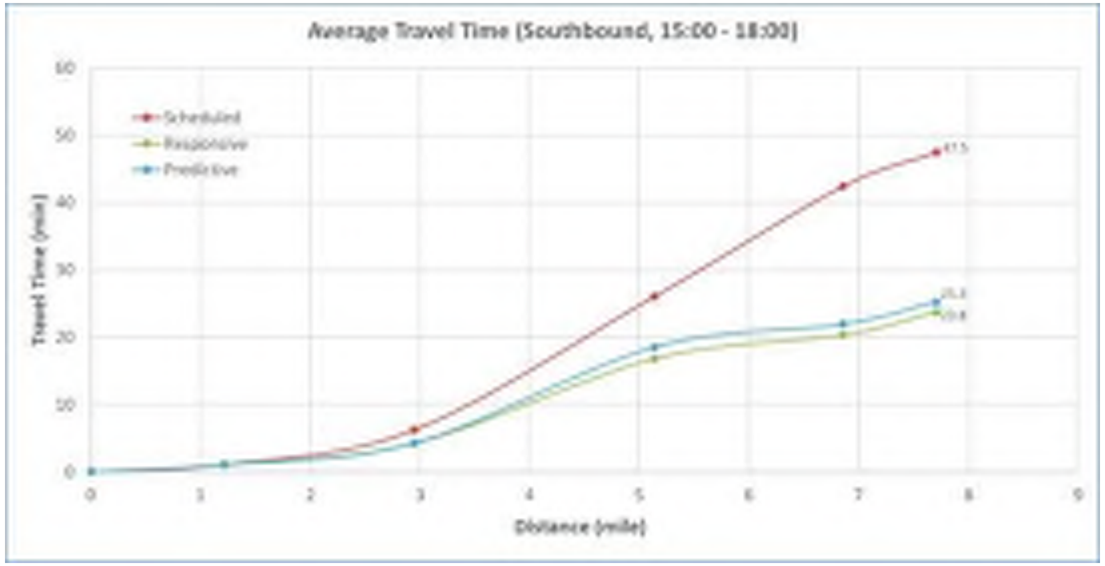


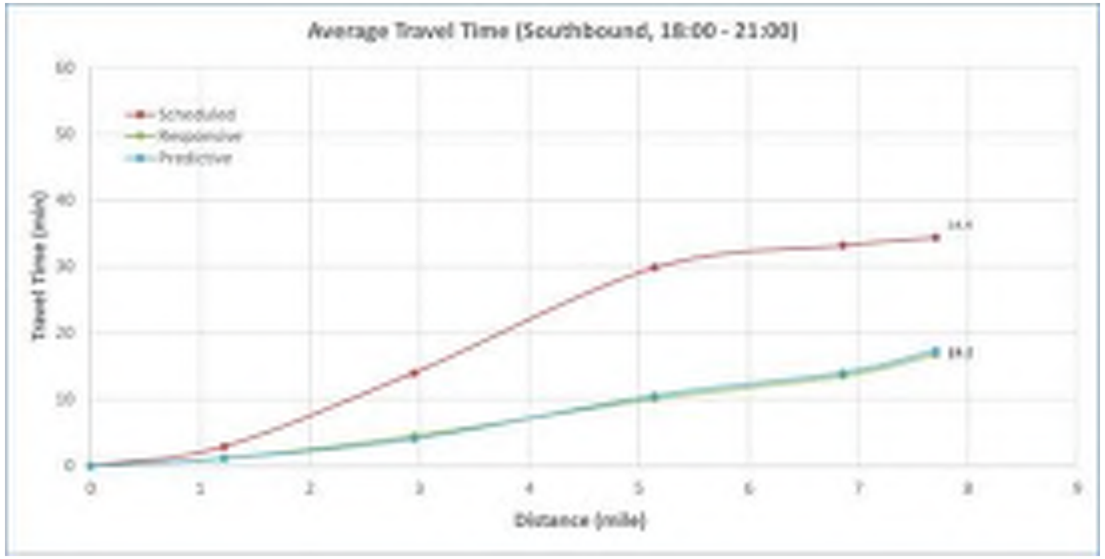


Travel Time Comparison - Saturday - Southbound

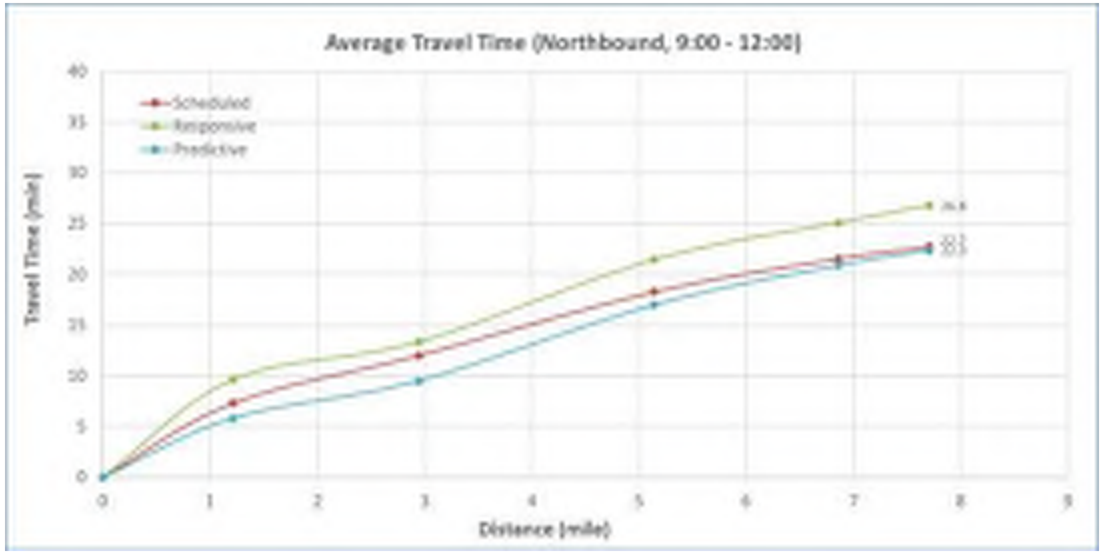


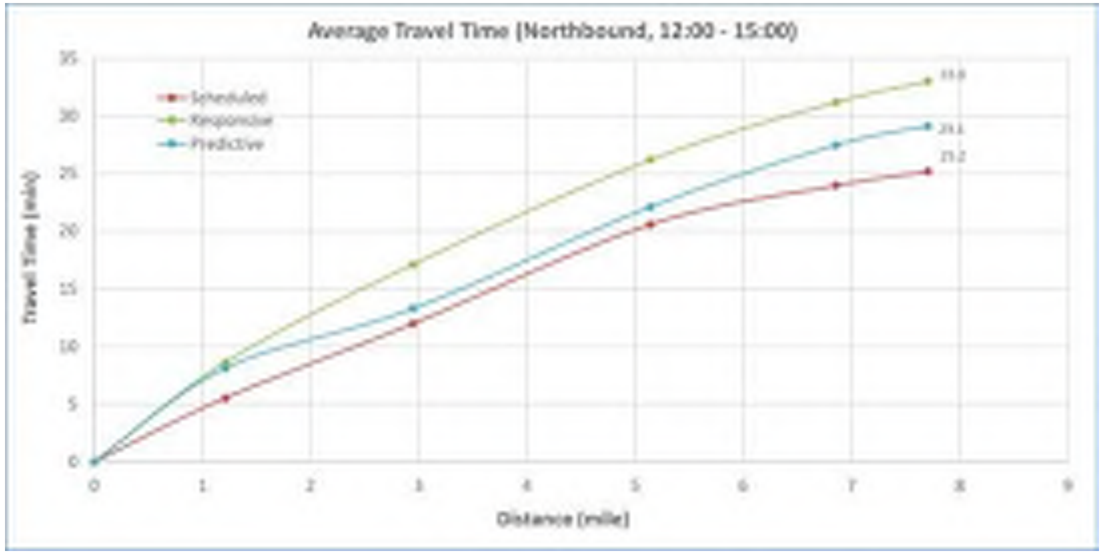


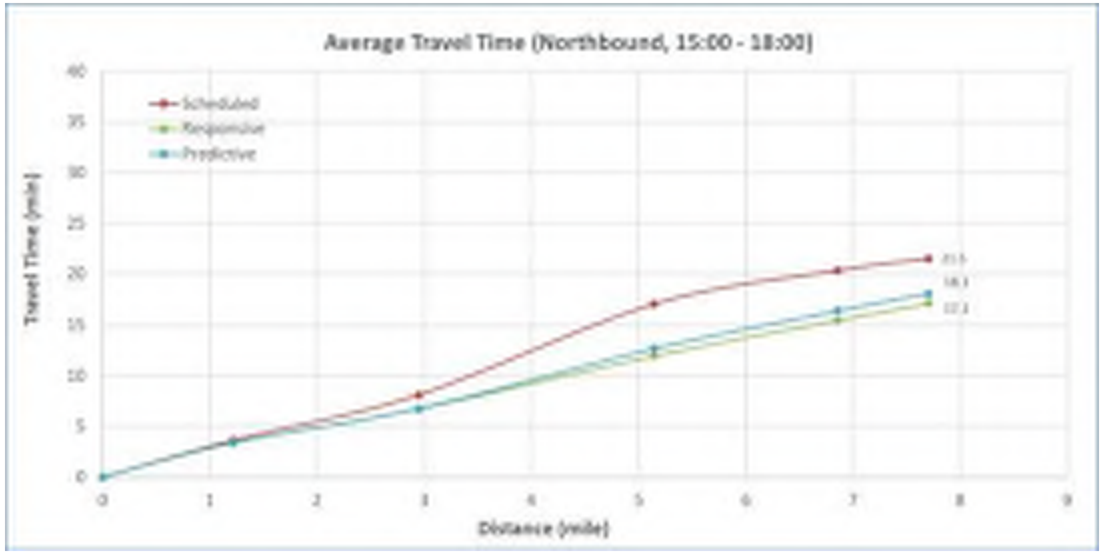


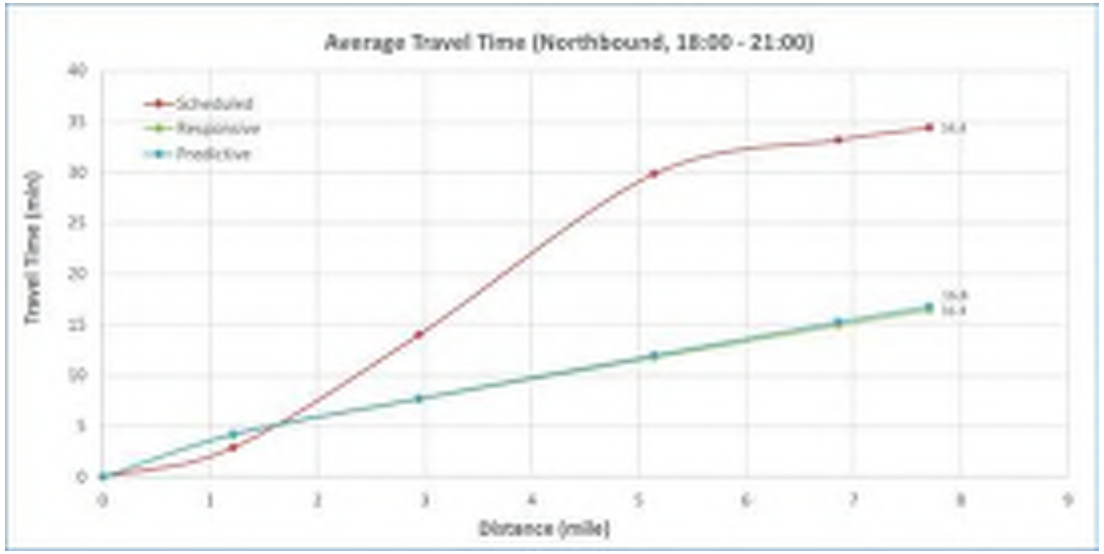


Travel Time Comparison - Saturday - Northbound









Summary

Conclusions & Next Steps

- Prediction Algorithms have the potential to improve arterial signal operations
- Provides proactive response, especially when demand is surging and uneven
- More testing underway
  - Varying scenarios such as incidents, work zones, special events etc.
  - Different time variant inputs can be trained and tested.
  - Multiple hybrid machine learning algorithms can be trained and tested.
- Limitation
  - Analysis limited by field-measured demand, which is dependent on field signal operation.
  - Pilot deployments planned in the future, they will provide a better assessment of benefits

Questions?

Zelan Jia, PE - Zelan.jia@jacobs.com

Joyce Zhou, PE, PTOE – Joyce.Zhou@jacobs.com

Shankar Natarajan, PE, PTOE - shankar.natarajan@jacobs.com