



Minnesota
Department of
Transportation

**RESEARCH
SERVICES
&
LIBRARY**

**Office of
Transportation
System
Management**

Expanding and Streamlining of RTMC Freeway Network Performance Reporting Methodologies and Tools

John Hourdos, Principal Investigator
Minnesota Traffic Observatory
Department of Civil Engineering
University of Minnesota

January 2014

Research Project
Final Report 2014-05



To request this document in an alternative format call [651-366-4718](tel:651-366-4718) or [1-800-657-3774](tel:1-800-657-3774) (Greater Minnesota) or email your request to ADArequest.dot@state.mn.us. Please request at least one week in advance.

Technical Report Documentation Page

1. Report No. 2014-05	2.	3. Recipients Accession No.	
4. Title and Subtitle Expanding and Streamlining of RTMC Freeway Network Performance Reporting Methodologies and Tools		5. Report Date January 2014	
		6.	
7. Author(s) John Hourdos and Stephen Zitzow		8. Performing Organization Report No.	
9. Performing Organization Name and Address University of Minnesota Minnesota Traffic Observatory Department of Civil Engineering 500 Pillsbury Drive SE Minneapolis, MN 55455		10. Project/Task/Work Unit No. CTS Project #2011036	
		11. Contract (C) or Grant (G) No. (C) 89261 (WO) 203	
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Research Services & Library 395 John Ireland Boulevard, MS 330 St. Paul, MN 55155		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes http://www.lrrb.org/pdf/201405.pdf			
16. Abstract (Limit: 250 words) <p>The Twin Cities freeway network is a densely instrumented and growing transportation system. As the Minnesota Department of Transportation (MnDOT) pursues a performance-based management strategy to monitor the health of the network and make planning and management decisions, the data from this vast network is being examined using a variety of methods. To provide MnDOT with timely performance information regarding the Twin Cities freeway network, a streamlined program was developed based on existing and new methodologies. The Highway Automated Reporting Tool (HART) utilizes a user-friendly interface for corridor, date, time, and report selection. Selected data are automatically examined to identify and correct errors and produce 'cleaned' data for use within each report. Using interpolation and autoregressive integrated moving average (ARIMA) techniques, small errors are corrected in place while preserving as much useful data as possible. Larger issues are corrected using an imputation algorithm relies on uses nearby sensors and historical data to create representative replacement data. In this first version of HART, four reports are included: Speed-Based Congestion, Congestion Intensity, Lost and Available Capacity, and Maximum and Total Throughput. The Speed-Based Congestion report matches the existing methodology used to generate the annual Congestion Report.</p>			
17. Document Analysis/Descriptors Traffic measurement, Traffic performance, Analysis		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Alexandria, VA 22312	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 106	22. Price

Expanding and Streamlining of RTMC Freeway Network Performance Reporting Methodologies and Tools

Final Report

Prepared by:

John Hourdos
Stephen Zitzow

Minnesota Traffic Observatory
Department of Civil Engineering
University of Minnesota

January 2014

Published by:

Minnesota Department of Transportation
Research Services & Library
395 John Ireland Boulevard, MS 330
St. Paul, MN 55155

This report represents the results of research conducted by the authors and does not necessarily represent the views or policies of the Minnesota Department of Transportation or the University of Minnesota. This report does not contain a standard or specified technique.

The authors, the Minnesota Department of Transportation, and the University of Minnesota, do not endorse products or manufacturers. Any trade or manufacturers' names that may appear herein do so solely because they are considered essential to this report.

Acknowledgements

We would like to thank the Minnesota Department of Transportation (MnDOT) for supporting this project. We would like to acknowledge the help we received from the Center for Transportation Studies in coordinating and managing this project. We would like to acknowledge the help, support, and cooperation of Mr. Brian Kary, Mr. Jesse Larson, and others at the MnDOT Regional Traffic Management Center. We would also like to thank the Minnesota Traffic Observatory for hosting this research project, providing material and software support.

Table of Contents

1. Introduction.....	1
2. User Interface Design	3
Corridor Selection (A).....	3
Corridor Selection Feedback Map (B)	4
Date Selection (C)	4
Time Selection and Other Options (D)	5
Report Selection (E).....	5
3. Data Cleaning and Imputation Module.....	6
DSE to DCI Module Overlap.....	6
Preliminary Data Cleaning.....	7
Reality Checks.....	7
Interpolation.....	9
Outlier Correction through ARIMA.....	9
Imputation of Broken Detectors.....	10
Sibling Selection.....	10
Neighbor Selection	12
Imputation Date Selection	13
Correlation.....	13
Imputation.....	16
4. Report Generation and Visualization.....	20
Speed-Based Congestion.....	20
Possible Improvement	26
Congestion Intensity.....	26
Lost Capacity Report Module	27
Possible Improvement	30
Available Capacity Report Module.....	30
Maximum Throughput and Total Throughput	33
Report Visualization.....	36
5. Discussion.....	38
6. Conclusion	39
References.....	40
Appendix A – Installation Instructions	
Appendix B – User Manual	
Appendix C – Imputation Figures	

List of Figures

Figure 1 Major components of HART	1
Figure 2 User interface.....	4
Figure 3 Data Selection and Extraction (DSE) and Data Cleaning and Imputation (DCI) module subcomponents	6
Figure 4 Data cleaning and storage prior to imputation	8
Figure 5 Data imputation for broken detectors	11
Figure 6 Sibling imputation component	12
Figure 7 Neighbor imputation component flow diagram	14
Figure 8 Data imputation process	16
Figure 9 Report generation modules in relation to the overall HART program.....	20
Figure 10 Speed-Based Congestion module methodology flow diagram	21
Figure 11 Speed-Based Congestion report visualization for AM-peak of October 2012 based on program cleaned data.....	22
Figure 12 Speed-Based Congestion report visualization for AM-peak of October 2012 published by MnDOT.....	23
Figure 13 Sample data for detector 6252 along I-694, flow indicated in the top plot and speed in the bottom plot	24
Figure 14 Sample data for detector 65 along I-94, flow indicated in the top plot and speed in the bottom plot	25
Figure 15 Congestion Intensity report for AM peak using October 2013 data	27
Figure 16 Lost Capacity module methodology flow diagram	28
Figure 17 Lost Capacity report visualization.....	29
Figure 18 Available Capacity module methodology flow diagram.....	31
Figure 19 Available Capacity report visualization	32
Figure 20 Maximum and Total Throughput modules methodology flow diagram	33
Figure 21 Total Throughput report for AM peak using October 2013 data	34
Figure 22 Max Throughput report for AM peak using October 2013 data	35
Figure 23 Flow diagram of Performance Measure Visualization module.....	36
Figure 24 Sample output from Speed Based Congestion Report	37

List of Tables

Table 1 Detectors and dates selected for testing imputation.....	17
Table 2 Squared error for volume data	18
Table 3 Squared error for occupancy data	18
Table 4 Squared error for speed data	19
Table 5 Average percentages by correlation.....	19

Executive Summary

The Twin Cities freeway network is a densely instrumented and growing transportation system. As the Minnesota Department of Transportation (MnDOT) pursues a performance-based management strategy to monitor the health of the network and make planning and management decisions, the data from this vast network is being examined using a variety of methods. To provide MnDOT with timely performance information regarding the Twin Cities freeway network, a streamlined program was developed based on existing and new methodologies. The Highway Automated Reporting Tool (HART) utilizes a user-friendly interface for corridor, date, time, and report selection. Selected data are automatically examined to identify and correct errors and produce 'cleaned' data for use within each report. Using interpolation and autoregressive integrated moving average (ARIMA) techniques, small errors are corrected in place while preserving as much useful data as possible. Larger issues are corrected using an imputation algorithm relies on nearby sensors and historical data to create representative replacement data.

In this first version of HART, four reports are included: Speed-Based Congestion, Congestion Intensity, Lost and Available Capacity, and Maximum and Total Throughput. The Speed-Based Congestion report matches the existing methodology used to generate the annual Congestion Report. Through HART, the congestion report can more easily and quickly be generated. The other reports extend the performance-based management toolbox available for MnDOT engineers and planners to better examine congestion, the behavior of bottleneck locations, and the potential benefits stemming from investments in the network. The Congestion Intensity report extends the congestion report to account for the number of lanes present across the network and highlights severe congestion at multi-lane segments of the system. The Lost and Available Capacity report identifies the amount of 'potential' that is lost during congested conditions and the spare capacity that is available for locations that do not experience breakdowns. This report, in particular, has the possibility to help direct planning and investment to locations across the system that have the most potential for efficient improvement. Finally, the Maximum and Total Throughput report shows the usage pattern across the network.

HART was developed to not only tie in with existing MnDOT methodologies and reporting structures but also to be extensible and modifiable. As HART is used to examine the Twin Cities system, new performance measures can be easily integrated into the program and existing methodologies can be augmented to more directly serve planning and management needs. Throughout these performance measure modifications, the core of HART's data cleaning and imputation algorithms can also be reexamined to improve these methodologies, resulting in increasingly accurate analyses of the Twin Cities freeway network.

1. Introduction

This report describes the program developed through the “Expanding and streamlining of RTMC freeway network performance reporting methodologies and tools” project. In accordance with the Minnesota Department of Transportation’s (MnDOT’s) performance based management strategy, the Regional Transportation Management Center (RTMC) developed report methodologies and data structures to examine the behavior of the network over time. However, those methods were performed manually and were time-intensive, requiring significant effort to monitor the performance of the network using only a sample of the significant quantity of data generated each year. The Minnesota Traffic Observatory was commissioned to develop a new automated tool that will assist in these tasks. The Highway Automated Report Tool (HART) was developed to reduce the time investment required to produce useful information regarding the network. HART also includes several new performance measures which offer new insights into the operation of the Twin Cities freeways.

HART can be broken down into three major components: Data Selection and Extraction, Data Cleaning and Imputation, and Report Generation and Visualization (see Figure 1).

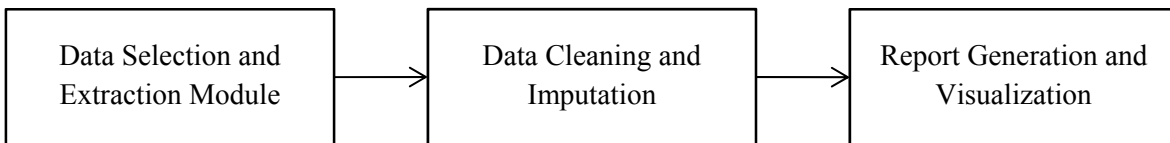


Figure 1 Major components of HART

The Data Selection and Extraction module includes the user interface which includes corridor, date, time, and report selection. A small map is also included which updates automatically to show which portions of the network are selected for data retrieval and analysis.

The Data Cleaning and Imputation module retrieves and cleans data prior to report generation. Each detector-day of data is cleaned independently to identify and fix several types of simple errors. This includes using an autoregressive integrated moving average (ARIMA) model to correct outlier data points. For major errors, entire days are imputed using nearby and historical detector data. Cleaned and imputed data is stored locally in a database.

Finally, the Report Generation and Visualization modules examine this cleaned data and generate information describing the performance of the network in several ways. Four reports are included within HART:

- Speed-Based Congestion
- Congestion Intensity
- Lost and Available Capacity
- Maximum and Total Throughput

The Speed-Based Congestion report matches the existing methodology set forth by the RTMC and focuses on recurring congestion throughout the network. The Congestion Intensity report is an extension of the Speed-Based Congestion report which accounts for the geometry of the system. The Lost and Available Capacity report examines the severity of congestion in terms of

unused or needed capacity. Finally, the Maximum and Total Throughput report maps the usage of the network at peak times and in aggregate.

Each report is tabulated within HART and written to a geospatial database (the same one used for the past RTMC methodology) which automatically plots a figure showing the calculated values across the network. The figures automatically update with each use of HART, dramatically reducing the time required to update and correct report visualizations.

The entire program was developed in Java to match the code base used by the RTMC. As such, the code underlying HART can be easily maintained and improved over time by MnDOT engineers and planners as new methodologies or performance measures are introduced. HART is an important first step toward integrating, automating, and improving the reporting tools and methodologies used to examine the Twin Cities freeway network.

2. User Interface Design

The Data Selection and Extraction (DSE) module forms the user interface of HART. The DSE is designed with functionality similar to the DataExtract tool in the way it recognizes the roadway network geometry and retrieves the data from the MnDOT loop-detector database. For the geometry HART utilizes the information stored in the metro_config.xml which is also the basis for the data structures of IRIS and other RTMC applications. HART includes code to process the older tms_config.xml files used prior to the redesign of IRIS which separated “Twin Cities Metro” and “District 1”. This file contains data for each detector and station such as lane numbering, type, speed limit, etc. as well as the connections between the stations.

Since HART is compatible with past versions of the configuration file, this gives flexibility to store configurations and process historical data with the settings that were present at the time. In particular, the settings for the detector will be relevant to the time period being studied (field length especially, but also speed limit and other attributes), but the overall structure of the network will also be correct (number of lanes, detectors belonging to each station, etc.). Examining historical data using a current-day configuration will cause poor estimations of speed, leading to erroneous results regarding congestion and lost and available capacity.

Data is extracted from the MnDOT database through Trafdat (Traffic Data Servlet). Queries are assembled for each detector-date required by the DSE module and sent to the Trafdat. Volume and occupancy data for the requested detector-date are returned and stored in a local database for future use. As these values enter HART, they are cleaned of simple errors and outliers. The details regarding the cleaning follow in the Data Cleaning and Imputation module section. The following subsections refer to regions of the user interface as shown in Figure 2.

Corridor Selection (A)

From the metro_config system structure all the corridors in the MnDOT freeway system were enumerated. Essentially “corridors” is a list of all the freeways separated by route (freeway) and direction (NB, SB, etc.). For each corridor the r_node structure contains the sequence of station upstream to downstream as well as the detector IDs comprising each station. Internally the user’s selection of corridor is translated into a list of detectors the module will extract from Trafdat. Only mainline detectors are included within this list. It is possible in the future that this portion of the UI will no longer cover the entire range of possible corridors/directions and will require modification. However, there are several corridors on the list which are not yet populated with detectors and are in place for future growth of the network.

The user is able to select each corridor individually by checking the box next to the name, select or unselect all with the two buttons provided, or import a custom corridor structure. The custom corridor option reads a text file containing a list of detector IDs (each starting with the letter ‘D’). These detectors are extracted and used throughout HART in the same way as any other corridor. No other corridors are selected when a custom corridor is loaded.

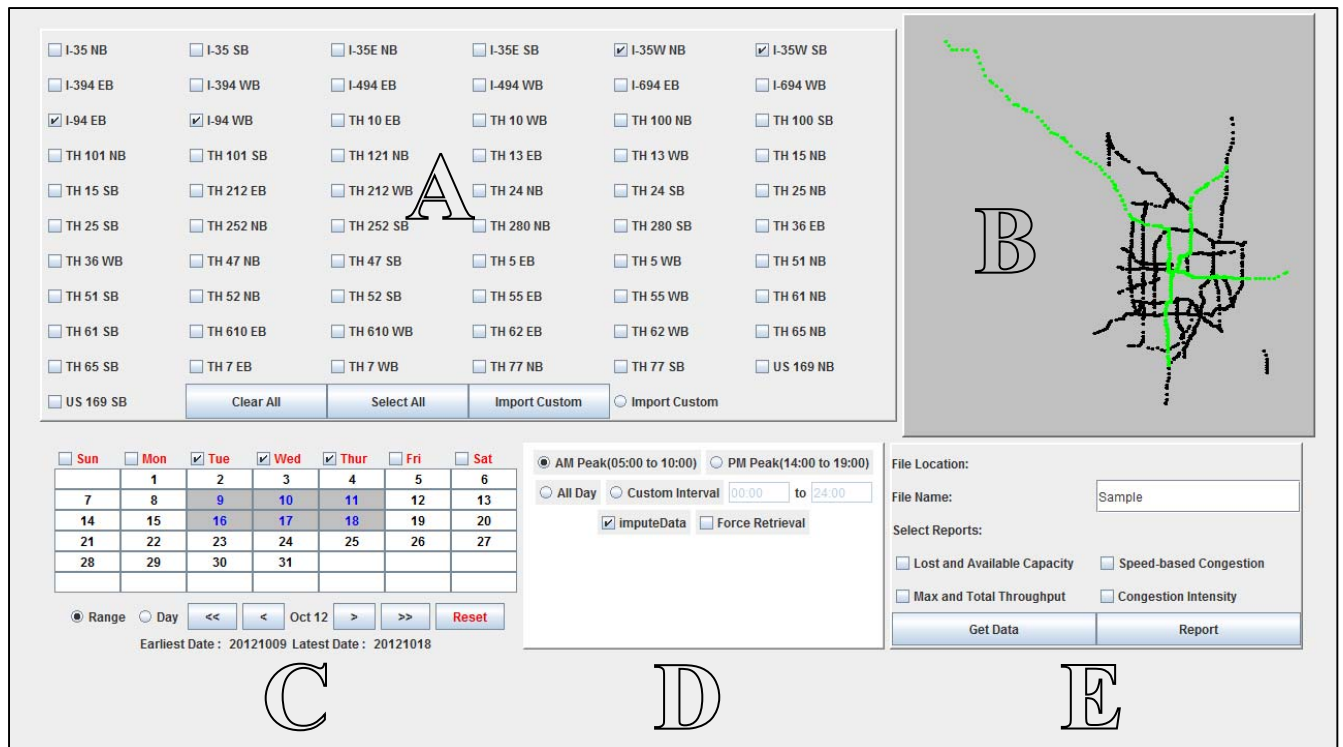


Figure 2 User interface

Corridor Selection Feedback Map (B)

In the area adjacent to the corridor selection region, a map of the system gives feedback to the user on the corridors selected. Each corridor is highlighted when selected. If a custom corridor is loaded, any nodes that have detectors in the list are highlighted in the map region.

Date Selection (C)

The day selection area is similar to the one found in DataExtract but in the case of DSE it offers more functionality. Specifically, it has two selection styles: “Range” and “Day”. In the “Range” selection style the user defines a range of days the report will be based on by selecting a start and end date. It is possible to select only certain days of the week from all the days inside a given range by toggling the check box for each day of the week. For example, keeping Tuesday, Wednesday, and Thursday toggled on and selecting June 1, 2011 followed by June 30, 2011 will highlight June 1-2, 7-9, 14-16, 21-23, and 28-30.

If no dates have been selected, clicking will select a date. If any dates have been selected, clicking will first flip the clicked date (on to off, or off to on) and then unselect any dates after the clicked date. Returning to the above example with the dates selected, clicking June 7th would leave only June 1-2 selected. Clicking May 31st instead would leave only May 31st selected.

In the “Day” selection style the functionality is the same as in DataExtract where the user manually selects each of the days he/she wants to run the report for. Each selected day is highlighted in the calendar view.

Using a combination of “Range” and “Day” selection styles, the user can select any number of days with any number of gaps between them. Regardless of gaps, the lower right side displays

the boundaries of the report period defined by the “Earliest Date” and the “Latest Date”. It is important to note that only toggled-on days of the week will be selectable in either mode. Clicking on untoggled days of the week will affect the dates selected if in “Range” mode, but will have no effect in “Day” mode. Toggling a day of the week on after a range has been selected will not retroactively select dates.

Time Selection and Other Options (D)

The time selection area allows the user to define the time of day the report(s) will cover. The selection time boundaries for the peak periods were based on a copy of a past congestion report and cover 5 AM to 10 AM and 2 PM to 7 PM. The “All Day” selection uses data for all 24-hours of each selected date. Finally, the user has the ability of defining an arbitrary custom time interval for the report. This interval will be used on all dates chosen in the date selection region. HART only allows one time interval selection at a time.

Two additional options are included in this region. “Impute Data” will either activate or deactivate the data imputation steps which normally follow cleaning. “Force Retrieval” will cause HART to ignore any data already present in the local database which may exist. All data retrieved will be ‘brand new’ from the Trafdat servlet and cleaned and imputed before being stored. If retrieval is not forced, the local database will be checked for existing data for the detector-dates selected. Any that are present will be used as-is. Retrieving the data from the MnDOT database is a relatively slow process which, if it can be avoided, greatly reduces the programs execution time.

The time selection window is used only when reports are generated. When data is retrieved, the entire day for each date selected will be retrieved, examined, and stored in the local database.

Report Selection (E)

This region contains two buttons which control the operation of HART: “Get Data” and “Report”. Before generating reports, data must be retrieved. “Get Data” will take the selected corridors and dates and retrieve 24-hour data. “Report” will account for the time selection and report selection and generate results. These values are updated into a user-selected report GIS database file for visualization.

Four report options are available for generation: Lost and Available Capacity, Speed-based Congestion, Max and Total Throughput, and Congestion Intensity. Any or all of these may be selected simultaneously and will be generated when the “Report” button is selected.

The “File Location” and “File Name” options are currently unused in HART. They were included from an earlier design of the program and could be used for simple data extraction but would require additional development.

3. Data Cleaning and Imputation Module

After selections are made in the Data Selection and Extraction (DSE) module, the Data Cleaning and Imputation (DCI) module checks extracted data for erroneous values and attempts to clean or impute values where possible. The cleaned and imputed data are then saved for the report generation modules of HART.

The process can be broken down into four steps: reality checking, interpolation, outlier correction using ARIMA, and imputation. The first three steps are performed on a detector-by-detector basis as data is requested from the Trafdat servlet. Data is then stored in a local database in a 'preliminary-clean' state. Any detector-dates requiring imputation are then processed again separately (if selected). Once imputation is completed for all necessary detectors, the data stored within the local database is considered 'clean' and ready for use in any report generation modules. Figure 3 below shows an outline of the process.

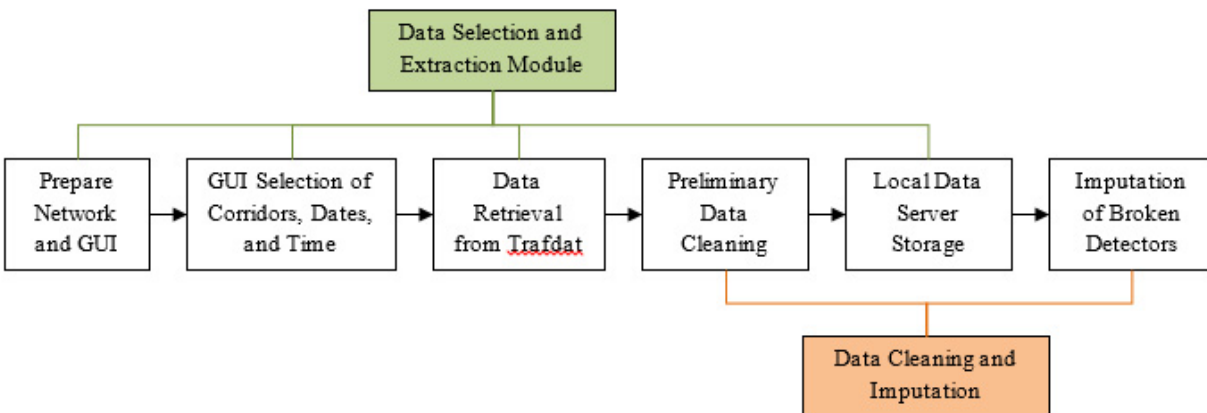


Figure 3 Data Selection and Extraction (DSE) and Data Cleaning and Imputation (DCI) module subcomponents

Throughout the process of cleaning, error flags are kept for each detector-date-time combination. Each error flag corresponds to one of the tests or steps below and serves as a history of the actions taken on each row.

DSE to DCI Module Overlap

As can be seen in Figure 3, the DSE and DCI modules are both involved in the data extraction and storage stages of HART. The first stages of the DCI module, namely data cleaning, interpolation, and outlier correction through ARIMA, are all performed on each detector-date independently. As such, those steps are handled as data comes from Trafdat and prior to data being stored in the local database. However, imputation relies on nearby detector information. To ensure that good data is used where possible, all selected data is cleaned before imputation begins.

DCI module can be broken into two halves: first, each detector-date is examined, checked for obvious bad data, missing data are marked, small holes are filled, and outliers are identified and corrected; second, each detector with large holes is checked for imputation and replaced if possible.

Preliminary Data Cleaning

Preliminary data cleaning consists of three steps: reality checks, interpolation of small holes, and correction of outliers using ARIMA filtering. Figure 4 below shows the pertinent portion of the DCI module. The sections that follow describe each step in detail.

Reality Checks

For each detector, four reality checks are made to identify any obviously nonsensical data. These four tests are:

Negative Values - Any negative values found within volume or occupancy

Out of Bounds Occupancy - Any occupancy values of greater than 100%

Occupancy with no Volume - Any time segment where volume is 0 and occupancy is more than 0%

Out of Bounds Speed - Any speed values greater than 100 mph

In cases of negative values, only the particular volume or occupancy values that are out of bounds are replaced with an error value. For the other three reality checks, all three columns (volume, occupancy, and speed) are replaced with error values.

It is important to note that requests to Trafdat are for a particular detector-date return 30-second volume and occupancy values, but not speed. The loop detectors in the network do not measure speed directly. Instead, speeds are estimated using volume, occupancy, and the calibrated field length of each detector. The raw data returned from Trafdat undergoes the first reality check (negative values) before speed is estimated.

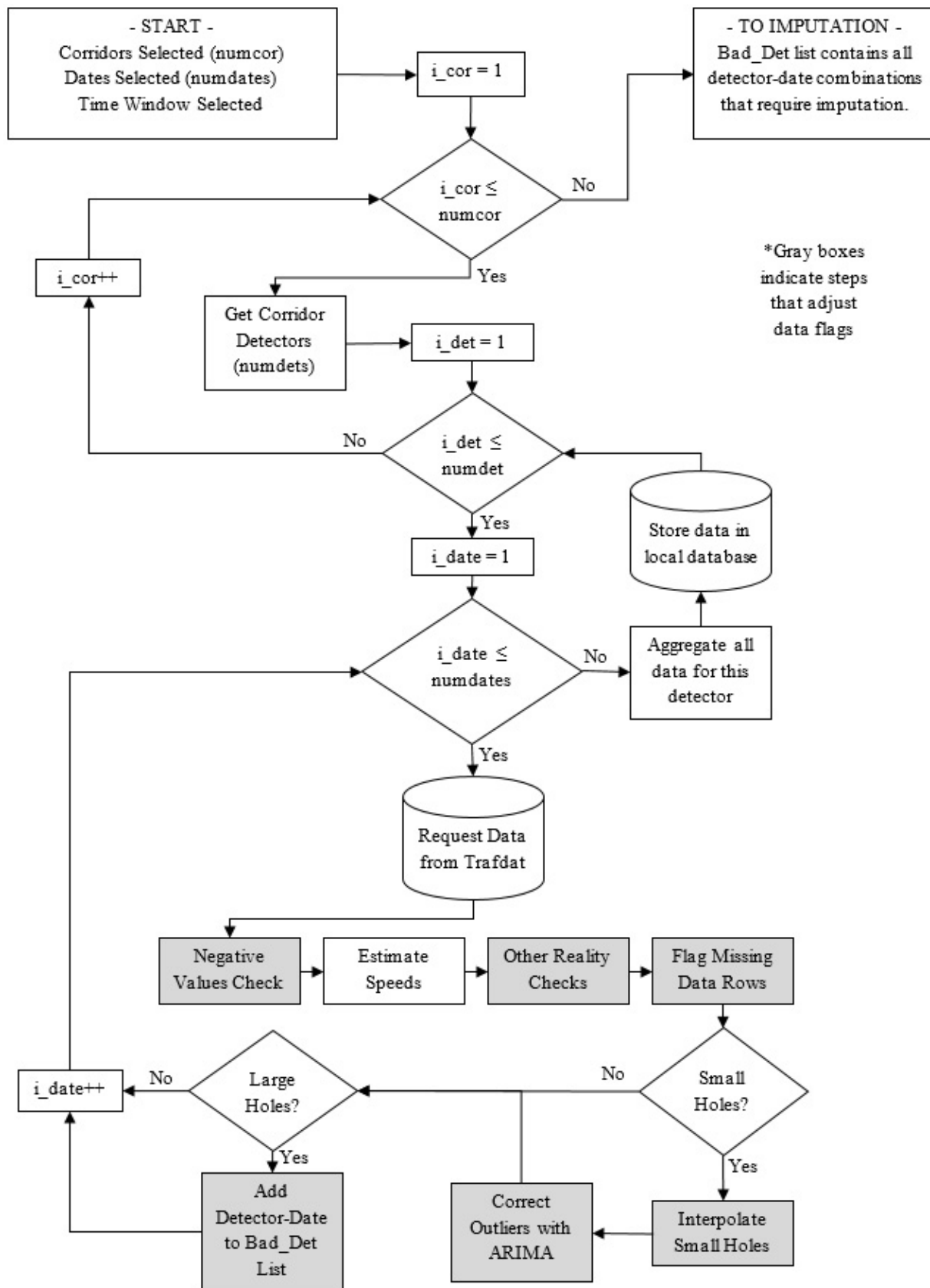


Figure 4 Data cleaning and storage prior to imputation

Interpolation

After reality checks are completed, any time segments that have missing data are identified. (It is possible that a row of data could contain missing data in one or two columns, but not all three. However, to be conservative, any row that has an error in any of the columns is considered to be faulty and the entire time segment is flagged as an error.)

In order to perform the next step, outlier correction using ARIMA, the data must be free of any holes or gaps (i.e. error values must be replaced). However, filling larger gaps with simple imputation introduces growing amounts of error. To minimize the error introduced, a limit is placed on the size of any given gap that can be imputed. For this tool, the largest hole that will be imputed is 5 consecutive 30-second intervals. Holes of 5 or fewer intervals are considered ‘small’ and any of 6 or more are considered ‘large’.

Small holes are interpolated based on whether they are internal or edge holes. Edge holes are those that are contiguous with either the start or end of the time period selected in the user interface. Any holes not contiguous with the edges are interior holes. Interior holes are interpolated using simple linear interpolation using surrounding good values, while edge holes are interpolated using nearest neighbor interpolation (i.e. the nearest good value is used to replace any missing values).

Outlier Correction through ARIMA

With the data set now consisting of large holes that will be imputed and regions of good data interspersed with interpolated values to fill holes, a statistical model is created to describe the data and check the non-hole regions for outlier values.

An autoregressive (AR) model of a series x_t implies that the series can be represented as the function of its past values. An AR model of order p , represented by AR (p), is formulated as: $x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \omega_t$, where, $\alpha = \mu(1 - \phi_1 - \dots - \phi_p)$, $\mu =$ mean of the series, and ω_t is assumed to be Gaussian white noise series with mean 0 and variance σ_w^2 . A moving average model of order q , or MA(q) of a time series x_t is defined as: $x_t = \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \dots + \theta_q \omega_{t-q}$. A time series is ARMA (Auto Regression Moving Average) or ARMA(p,q) if it is stationary and can be expressed as: $x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \dots + \theta_q \omega_{t-q}$

A process is said to be Auto Regression Integrated Moving Average, ARIMA (p,d,q) if ARMA(p,q) model is performed after differencing the series x_t with x_{t-d} (x_t lagged by d units). Differencing is done to make the series mean stationary as required by ARMA model.

In this project, ARIMA(1,0,1) was selected based on experimentation using varying p , d , and q values. This resultant model can be expressed mathematically as:

$$x_t = \alpha + \phi_1 x_{t-1} + \omega_t + \theta_1 \omega_{t-1}$$

For the volume, occupancy, and speed data for each detector, ARIMA is applied and model values are determined. Each model point is compared to its corresponding real value and a Z value is calculated. Any points that fall outside three standard deviations from the model value are considered outliers and are corrected to the model value. This preserves the majority of the

normal variation in the detector data and minimizes false outlier detection but still isolates extreme outlier values.

This method requires data that is stationary, whereas detector data is non-stationary. In order to make use of ARIMA for HART the data must be made stationary through a reversible process. The process is two-step: first, take the square root of each data point; second, find the difference between successive data points. It is important to note that the stationary data series has one less row than the original detector output it is representing. After outliers of the stationary series are detected and corrected, the first value from the original data is used to return the difference series to non-stationary, useful values.

Imputation of Broken Detectors

When detector data are missing for large segments of time, imputation is used to approximate the behavior of the detector. In order to impute a broken detector, a nearby detector must be functioning properly for the same time window and both the target and nearby detector must be functioning properly on some historical date. The imputation routines of the DCI search for usable nearby detectors from two groups: first, sibling detectors (also referred to as adjacent detectors) are checked; second, neighbor detectors are checked. Both of these groups will be discussed in the sections below. A flowchart illustrating the imputation portion of the DCI module is shown in Figure 5.

Sibling Selection

For each broken detector-date, the DCI module first checks any detectors at the same station as the target detector. Since these detectors are in close physical proximity, they are likely to exhibit higher correlation factors than detectors that are thousands of feet away. Mainline detectors can be of two types: standard or HOV. Only detectors of the same type as the target are used as possible imputation partners. For example, assume a station has 3 lanes where lane 1 is HOV and lanes 2 and 3 are standard. Lanes 2 and 3 could be imputation partners to each other should one of them break, but the HOV lane would be unable to use lanes 2 or 3 as they are of a different type. Lane 1 would require neighbor imputation as discussed below.

When imputing, it is desirable to select the nearest lane to the target so that traffic patterns between the two lanes are as similar as possible. If multiple siblings exist as imputation partners for the target detector, the one with the closest lane number will be chosen. If there is a tie, the lower lane number is selected as the tie-breaker.

If a sibling is selected as a possible partner, other dates must be checked for correlated traffic patterns from which to impute. The methods used for selecting possible dates, checking correlation values, and imputation are described in later sections. Figure 6 shows the steps that represent the 'Attempt Sibling Imputation' process.

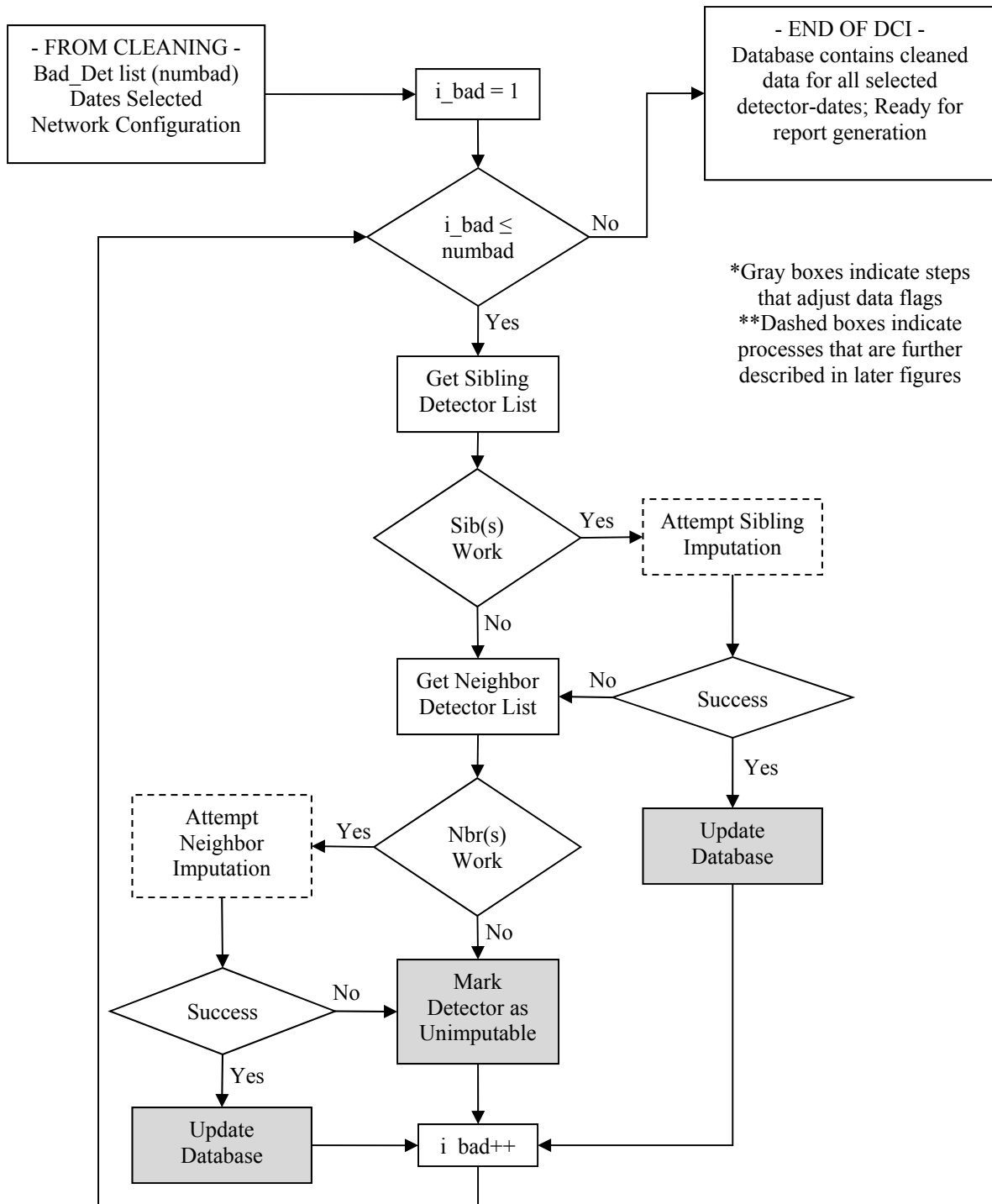


Figure 5 Data imputation for broken detectors

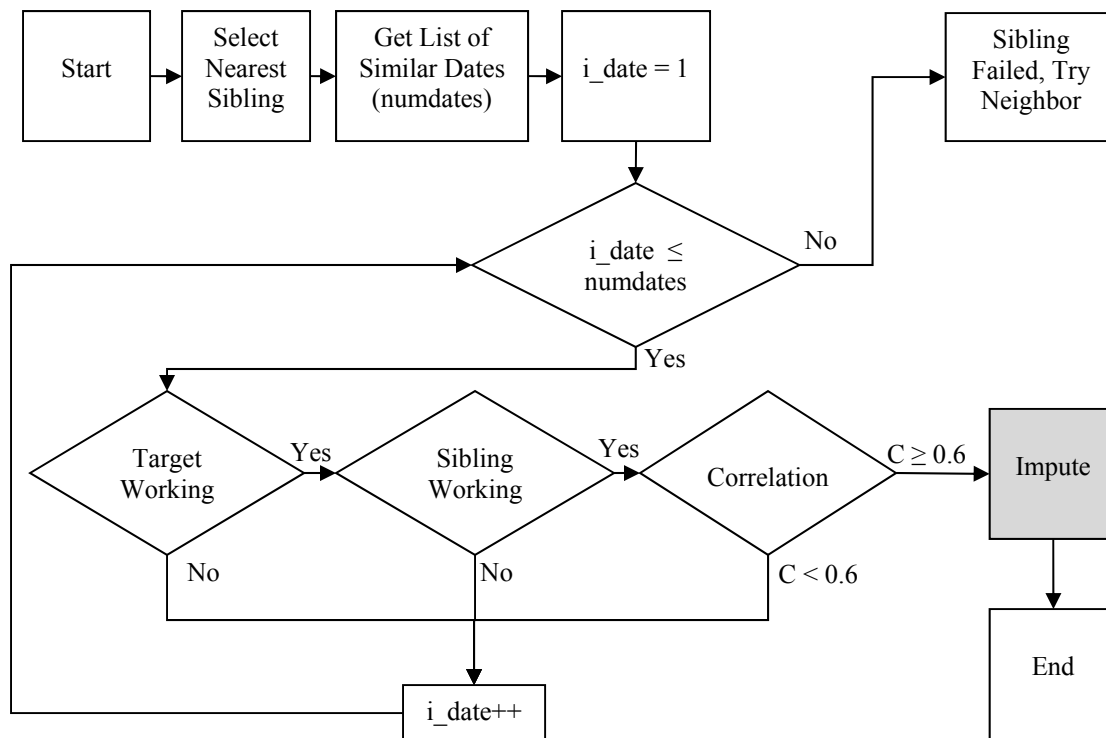


Figure 6 Sibling imputation component

Neighbor Selection

If no usable siblings exist for a target detector, neighboring detectors may be used to attempt imputation. Neighbor detectors are those found at the nearest upstream and nearest downstream stations. These detectors are not as closely correlated to the target as siblings would be, but may have sufficient correlation to give reasonable imputation results.

Ideally, the detector from the same continuous lane as the target detector would be selected both upstream and downstream. However, the network of the Twin Cities is large and has many lane additions and lane drops between stations that sometimes occur on the left and sometimes on the right. To simplify the neighbor search routine, all detectors of the same type (mainline or HOV) from both the upstream and downstream nearest stations are kept as possible imputation partners.

Returning to the example from before, suppose that the three lane segment mentioned is station S101. Upstream is station S100 with an HOV lane and a single standard lane. A lane enters and matches up to one of the regular lanes at station S101. Downstream is station S102 with three lanes, all three of which are standard. If the HOV in station S101 is broken, sibling imputation will be unavailable (no siblings of the same type), and a neighbor search will return the HOV from S100 as a possible partner. Since no HOV lane exists downstream, no downstream option will be returned. If, however, a standard detector is broken at station S101, four lanes will return as possible options: three from station S102, and one from S100.

Unlike sibling imputation, all of the possible neighbor partners will be carried forward into correlation analysis. The details of selecting the best option will be discussed in the section on

correlation. Figure 7 illustrates the steps that represent the ‘Attempt Neighbor Imputation’ process.

Imputation Date Selection

If a target detector has a functional sibling or neighbor detector on the date of interest, a historical date must be located where both detectors are working in order to impute. However, only dates that are similar to the missing date are considered for imputation. ‘Similar’ in the context of this tool means dates that are the same day-of-the-week, with the exception of Tuesday, Wednesday, and Thursday which are considered interchangeable.

Additionally, only certain date windows are considered. First, dates within the selected date list are checked. Second, dates not already selected but between the first and last date selected are checked. Third, dates up to 90 days prior to the first selected date are checked. Finally, dates one year previous to the selected are considered. This last window is made up of 90 days centered on one year previous to the middle of the selected dates. (For example, suppose that October 7-9th are selected this year. This last window would consider 45 days before and after October 8th of last year.)

These four windows are considered in the order listed above in order to find the nearest matches possible before considering less likely candidates.

Correlation

With the list of dates as described in the previous section, detector data must be examined and correlated. Only detectors that have no large holes within the selected time of day are allowed to be imputation partners. The target detector is checked for large holes first. If the target detector passes, each neighbor candidate is checked and any that are free of large holes are checked for correlation.

Most of the detector-date combinations that must be examined for correlation are not among those already selected and cleaned. For each detector-date combination, the DCI module first attempts to take data from the already cleaned local database. If sufficient data does not exist locally, additional calls to the Traftdat servlet are made to request additional data. These additional dates are subjected to preliminary cleaning and, if no large holes are found, interpolation and outlier correction with ARIMA. This cleaned data is then considered for correlation.

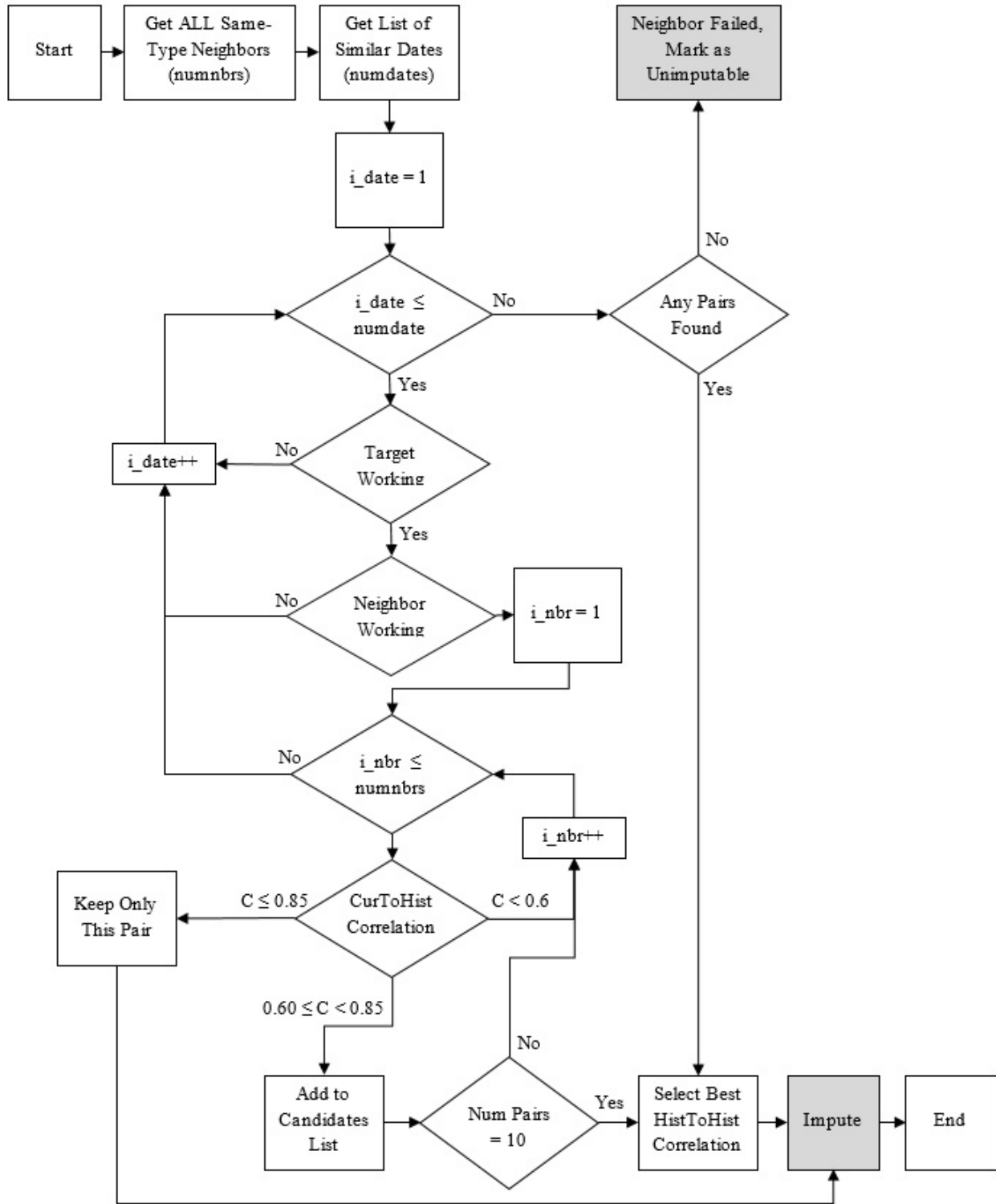


Figure 7 Neighbor imputation component flow diagram

Correlation is checked using a cross correlation technique. The loop-detector data used by HART are collected at equal intervals of time and can be referred to as time series data. In a time series, adjacent observations are inherently correlated which severely restricts the applicability of many traditional statistical methods. This is because many traditional models require adjacent

observations to be independent and identically distributed. Consequently, time series must be modeled and analyzed using techniques where the value of a time series at a given instance is modeled as a function of (1) different instance(s) of the same time series, (2) different or same instance of an independent time series, and (3) error.

Given two time series x_t and y_t , the cross-covariance function is given by:

$$\gamma_{xy}(s, t) = \text{cov}(x_s, y_t) = E[(x_s - \mu_{x_s})(y_t - \mu_{y_t})]$$

Where, $E[\dots]$ implies expectation function and s and t signifies two time instances (s and t may be identical). This can be scaled to another function called cross-correlation function defined as:

$$\rho_{xy}(s, t) = \frac{\gamma_{xy}(s, t)}{\sqrt{\gamma_x(s, s)\gamma_y(t, t)}}$$

Cross-correlation can be evaluated at different time instances of the two series, x_t and y_t . Lag is defined as the difference between s and t . The value of lag for which the two series have best alignment will give rise to maximum cross-correlation between the two series. Identifying this lag may be useful in deriving/modeling one series from/with the other. In this case, the lag can account for, as an example, the different onset times of congestion at a particular detector between two historical dates. A cross-correlation value of 1 implies that the two series are identical at a given lag. If two series have maximum cross-correlation at lag zero, they are identical series.

The first set of correlation factors is between the neighbor detector on the current date and the neighbor detector on the historical date (indicated by ‘CurToHist Correlation’ in Figure 7). For each pair examined, the occupancy time series is compared for correlation. The lag with the best correlation value is saved.

Even though it is mathematically possible to have the best correlation be at extremely large or small lags, only lags within 60 offsets of 0 are considered for HART. Traffic patterns may be slightly different between two dates, but are not likely to be more than a half hour earlier or later than any other date.

For sibling imputation, if any partner is found to have a correlation value of 0.60 or greater it is immediately used to impute the detector. Alternately, neighbor imputation has two stages. If a detector is found to have a correlation value of 0.85 or greater, it is immediately used, just as with sibling imputation. If, however, the correlation value is between 0.60 and 0.85 the partner is stored as a possible candidate. Up to 10 candidates in this range will be saved.

Once all the dates have been checked or if 10 of these candidates are found, a second round of correlation is performed. These correlations are between the target detector on the historical date and the neighbor detector on the historical date (indicated by ‘HistToHist Correlation’ in Figure 7). These correlations are to determine which detector upstream or downstream is most like the target, which is likely to indicate that the neighbor is in the same continuous lane either upstream or downstream. The highest of these correlation values is used to determine which partner is sent on to imputation

Imputation

Once a suitable candidate has been located using the correlation methodologies, data must be created to replace the unusable data at the target detector. This process involves three steps (shown in Figure 8 **Error! Reference source not found.** below):

- 1 – Perform regressions between usable data
- 2 – Apply regression factors to create intermediate imputed data
- 3 – Apply a blending factor to the two intermediate data sets to create final imputed data

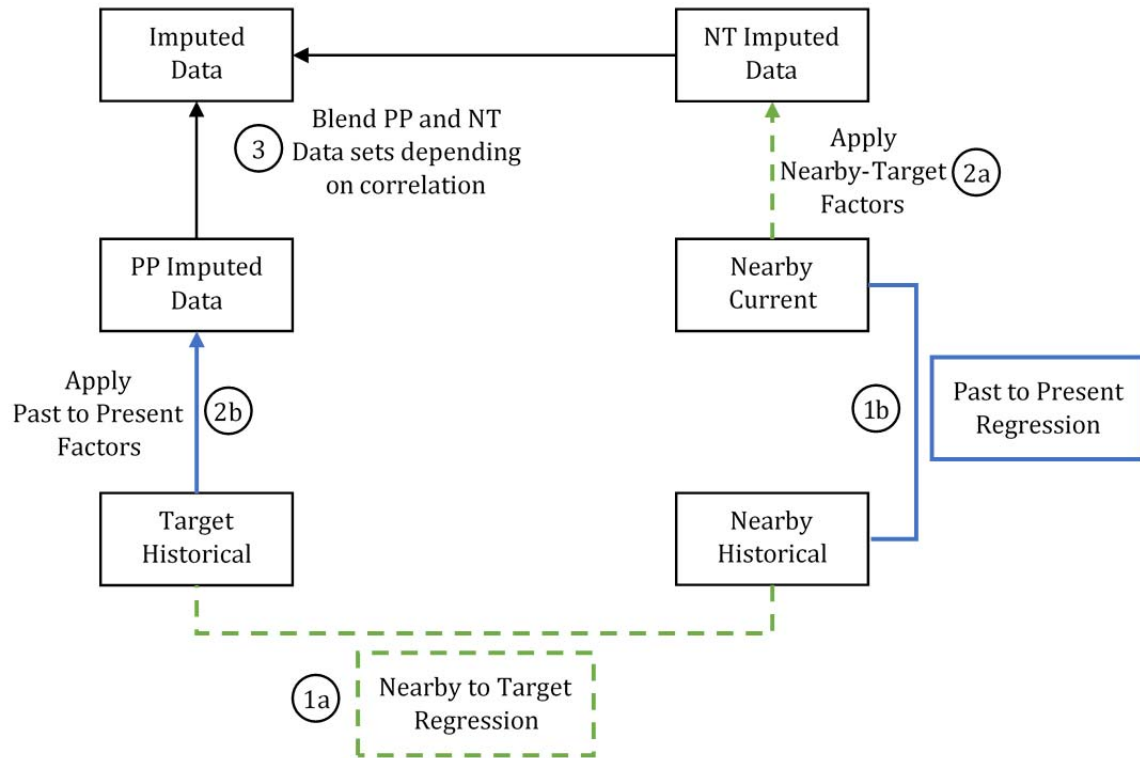


Figure 8 Data imputation process

Two regressions are performed using the usable data identified by the candidate selection process: between the nearby detector's historical data and the target detector's historical data (1a), and between the nearby detector's historical data and current data (1b). These are least-square linear regressions which return a constant factor and scaling factor representing the relationship between each pair of time series (volume, occupancy, and speed). These factors are then applied to create intermediate imputed data sets. The nearby-target factors are applied to the nearby detector's current data to approximate data for the broken detector. Similarly, the past-present factors are applied to the target detector's historical data.

Finally, the two intermediate data sets are blended based on the correlation value from the candidate selection process. If the correlation factor is less than 0.75, only the nearby-target imputed data is used (100%-0% blend), while if the correlation factor is more than 0.75 the nearby-target and past-present imputed data sets are each multiplied by one-half and summed (50%-50% blend).

This blending process is based on empirical tests performed using good detector data. Multiple sets of detectors with usable data were identified and one was marked as ‘bad’ and intermediate imputed data was generated. The imputed data was then compared to the real data using summed squared errors. In this way, three different blends of the intermediate data sets were compared to the real data to determine which blend best captures the real data’s behavior. Table 1 below shows the selected detectors, dates, and correlation factors. Table 2, Table 3, Table 4 show the squared error for volume, occupancy, and speed, respectively. Each value is paired with a percentage relative to the *best* match of the three for that time series. For example, in set #1, the 50-50 blend of volume data has the best squared error (4454.9) and so is marked with 100%. The next best, the Nearby-Target imputation only, has a value of 4898.9 which is 110% of the best (4454.9). The best matches are marked in green, the second best are marked in yellow, and the worst matches are marked in red. Data sets with correlations below 0.75 have a gray background.

Table 1 Detectors and dates selected for testing imputation

	#	Target	Nearby	Current Date	Historical Date	Correlation
AM Peak	1	339	340	10/5/2010	10/21/2010	0.802
	2	339	340	10/26/2010	10/27/2010	0.941
	3	512	513	10/13/2010	10/28/2010	0.913
	4	843	844	10/12/2010	10/28/2010	0.889
	5	1610	1611	10/12/2010	10/27/2010	0.838
	6	1610	1611	10/14/2010	10/27/2010	0.883
	7	2669	2670	10/14/2010	10/21/2010	0.936
	8	2670	2669	10/19/2010	10/28/2010	0.890
	9	3190	3189	10/21/2010	10/27/2010	0.922
	10	3235	3236	10/26/2010	9/30/2010	0.819
PM Peak	11	340	339	10/20/2010	10/21/2010	0.758
	12	513	512	10/21/2010	10/14/2010	0.887
	13	845	844	10/5/2010	10/12/2010	0.612
	14	1609	1610	10/13/2010	10/20/2010	0.677
	15	2646	2645	10/26/2010	10/20/2010	0.681
	16	2711	2712	10/28/2010	10/20/2010	0.734
	17	4109	4110	10/27/2010	10/26/2010	0.687

Table 2 Squared error for volume data

	#	Volume Squared Error		
		Past-Present	Near-Target	Blend
AM Peak	1	7337.9 (165%)	4898.9 (110%)	4454.9 (100%)
	2	7899.1 (163%)	5363.2 (111%)	4845.2 (100%)
	3	8666.8 (163%)	5909.0 (111%)	5328.5 (100%)
	4	9223.4 (139%)	8610.9 (130%)	6635.7 (100%)
	5	7241.3 (226%)	3210.2 (100%)	3531.8 (110%)
	6	6804.9 (189%)	3608.4 (100%)	3788.7 (105%)
	7	9672.0 (201%)	4816.3 (100%)	5216.4 (108%)
	8	11721.7 (155%)	9715.6 (128%)	7585.2 (100%)
	9	6119.3 (185%)	3299.2 (100%)	3426.8 (104%)
	10	5882.3 (168%)	3889.3 (111%)	3493.3 (100%)
PM Peak	11	6919.3 (144%)	4816.7 (100%)	5296.2 (110%)
	12	5258.7 (121%)	5317.4 (122%)	4346.4 (100%)
	13	6042.6 (122%)	5949.2 (120%)	4937.9 (100%)
	14	9233.7 (123%)	10249.2 (136%)	7532.0 (100%)
	15	7053.8 (158%)	4476.1 (100%)	4459.1 (100%)
	16	8969.4 (204%)	4395.6 (100%)	5622.7 (128%)
	17	9708.5 (173%)	5607.3 (100%)	6306.6 (112%)

Table 3 Squared error for occupancy data

	#	Occupancy Squared Error		
		Past-Present	Near-Target	Blend
AM Peak	1	56296.0 (218%)	31452.5 (122%)	25783.9 (100%)
	2	18320.0 (171%)	11300.0 (105%)	10713.9 (100%)
	3	54002.8 (235%)	22963.6 (100%)	26102.9 (114%)
	4	16991.9 (133%)	16350.2 (128%)	12752.4 (100%)
	5	49495.3 (185%)	27236.7 (102%)	26789.5 (100%)
	6	111011.4 (136%)	90656.2 (111%)	81834.5 (100%)
	7	41310.5 (205%)	20151.8 (100%)	21670.1 (108%)
	8	30175.7 (346%)	8719.1 (100%)	12132.3 (139%)
	9	7360.2 (184%)	4067.1 (102%)	4000.8 (100%)
	10	91206.2 (164%)	63392.5 (114%)	55733.0 (100%)
PM Peak	11	27345.8 (197%)	13869.3 (100%)	14297.2 (103%)
	12	41724.9 (154%)	43291.4 (159%)	27160.1 (100%)
	13	10145.1 (171%)	6479.7 (109%)	5937.6 (100%)
	14	85412.5 (154%)	68145.3 (123%)	55480.2 (100%)
	15	165553.1 (212%)	78167.5 (100%)	88764.9 (114%)
	16	137924.2 (202%)	68164.2 (100%)	75636.9 (111%)
	17	6605.1 (153%)	6008.1 (139%)	4313.6 (100%)

Table 4 Squared error for speed data

	#	Speed Squared Error					
		Past-Present		Near-Target		Blend	
AM Peak	1	97896.8	(188%)	52711.8	(101%)	52128.0	(100%)
	2	82190.7	(135%)	68064.4	(111%)	61107.6	(100%)
	3	119579.2	(152%)	100945.9	(128%)	78662.9	(100%)
	4	73976.1	(111%)	87323.0	(131%)	66772.4	(100%)
	5	173455.8	(162%)	114068.1	(107%)	107028.1	(100%)
	6	241311.4	(193%)	125029.9	(100%)	144361.1	(115%)
	7	20063.1	(175%)	11449.1	(100%)	12508.5	(109%)
	8	31563.8	(122%)	41210.9	(159%)	25978.1	(100%)
	9	36650.7	(100%)	39489.1	(108%)	37698.5	(103%)
	10	293795.9	(121%)	243677.0	(100%)	251002.8	(103%)
PM Peak	11	43926.0	(157%)	32008.4	(115%)	27928.7	(100%)
	12	102419.4	(158%)	72691.3	(112%)	65011.0	(100%)
	13	66913.7	(176%)	44274.4	(117%)	37975.5	(100%)
	14	140706.2	(207%)	72126.2	(106%)	67867.5	(100%)
	15	176205.6	(615%)	28647.8	(100%)	57360.5	(200%)
	16	195747.5	(857%)	22849.0	(100%)	65171.0	(285%)
	17	40079.9	(122%)	36117.0	(110%)	32897.7	(100%)

Based on the percentages listed in the figures above, the average was calculated for each time series and blend. These averages are listed in Table 5 below, along with averages for series with correlations above and below 0.75.

Table 5 Average percentages by correlation

	Volume Squared Error			Occupancy Squared Error			Speed Squared Error		
	Past-Present	Near-Target	Blend	Past-Present	Near-Target	Blend	Past-Present	Near-Target	Blend
Average Percent, All Samples	(165%)	(111%)	(105%)	(189%)	(113%)	(105%)	(221%)	(112%)	(119%)
Average Percent, Correlation ≥ 0.75	(168%)	(110%)	(103%)	(194%)	(112%)	(105%)	(148%)	(114%)	(103%)
Average Percent, Correlation < 0.75	(156%)	(111%)	(108%)	(178%)	(114%)	(105%)	(395%)	(107%)	(157%)

From these values, it can be seen that the Nearby-Target and 50-50 Blend are much better than the Past-Present imputation alone. However, the poor performance of the Blend can be seen for the Speed time series when correlation is under 0.75 (157%). As such, the Blend is used data sets with a correlation of 0.75 and above, while the Near-Target, which is only slightly worse for Volume and Occupancy, is used for the lower correlation sets.

Figures of each of the sample data sets can be found in Appendix C.

4. Report Generation and Visualization

To make use of the cleaned and imputed data generated by the preceding modules, there are six different reports included within HART:

- 1 – Speed-Based Congestion
- 2 – Congestion Intensity
- 3 – Lost Capacity
- 4 – Available Capacity
- 5 – Maximum Throughput
- 6 – Total Throughput

These six reports are grouped into three pairs based on the information that is processed and the methodologies used. The Speed-Based Congestion and Congestion Intensity reports, Lost and Available Capacity reports, and Maximum and Total Throughput reports are each closely related. The report modules rely on the data extracted and cleaned by the previous modules, and output information ready for integration with visualization tools. Figure 9 below shows the position of these reports in the overall tool structure.

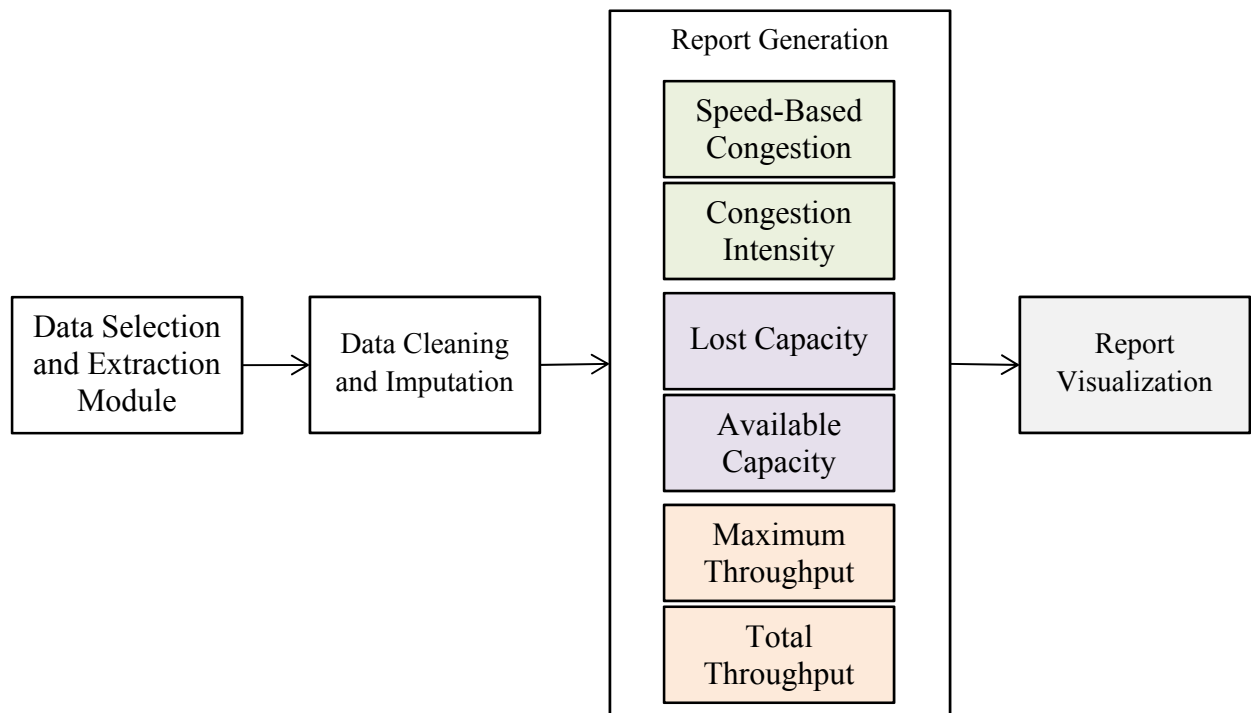


Figure 9 Report generation modules in relation to the overall HART program

Speed-Based Congestion

The Speed-Based Congestion (SBC) module follows the same methodology as the existing congestion report created by MnDOT (based on the queries and development flow found within the report generation database from 2010). The SBC report focuses on determining the duration of congestion at any location in the network and thereby identifying the most severely congested locations. On a larger scale, the SBC report also gives a measure on the overall health of the network from corridor to corridor and over time. The current Congestion Report developed by

the RTMC is created annually, but with HART additional reports could be generated at shorter intervals to give a more granular view of congestion patterns across the network.

Figure 10 below shows the flow diagram that describes the methodology. The report is generated on a station-by-station basis for all selected corridors. For every 5-minute interval during AM peak, PM peak, or other specified time interval, the 30-second detector data is aggregated (average speed, not weighted by volume). For every 5-minute segment, the median value of speed (across all selected days) is kept to form a ‘median day’. Finally, the median day detector measurements in each station are compared across each 5-minute interval and the minimum speed is selected as the station speed. These median day minimum station speeds are used to tabulate the speed-based congestion based on the 45-mph threshold used currently.

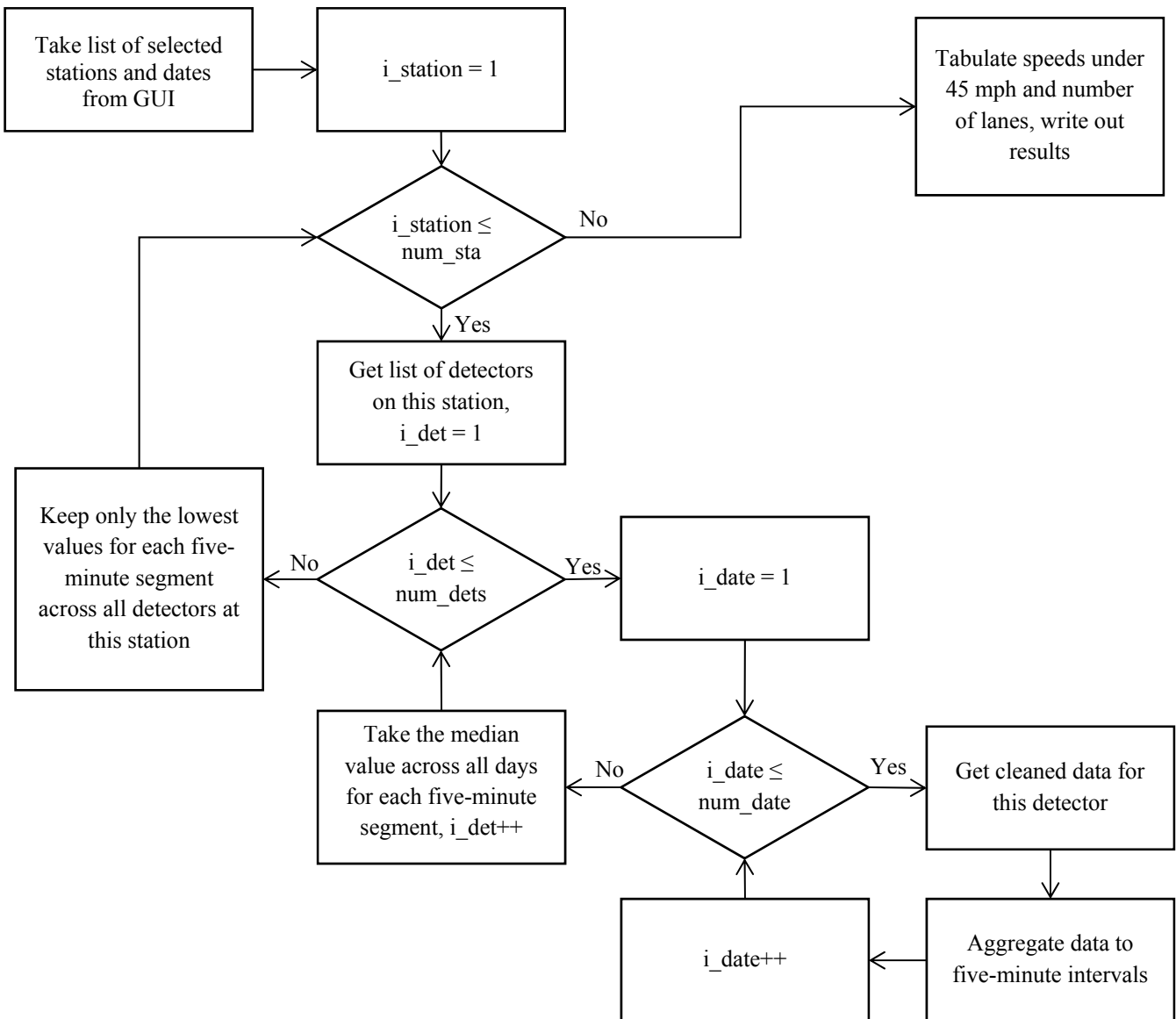


Figure 10 Speed-Based Congestion module methodology flow diagram

Figure 11 shows the visualization of the results for AM-peak for the midweek days between October 2nd and 25th, 2012. For reference, Figure 12 shows the visualization of the results from the 2012 Congestion Report published in January 2013.

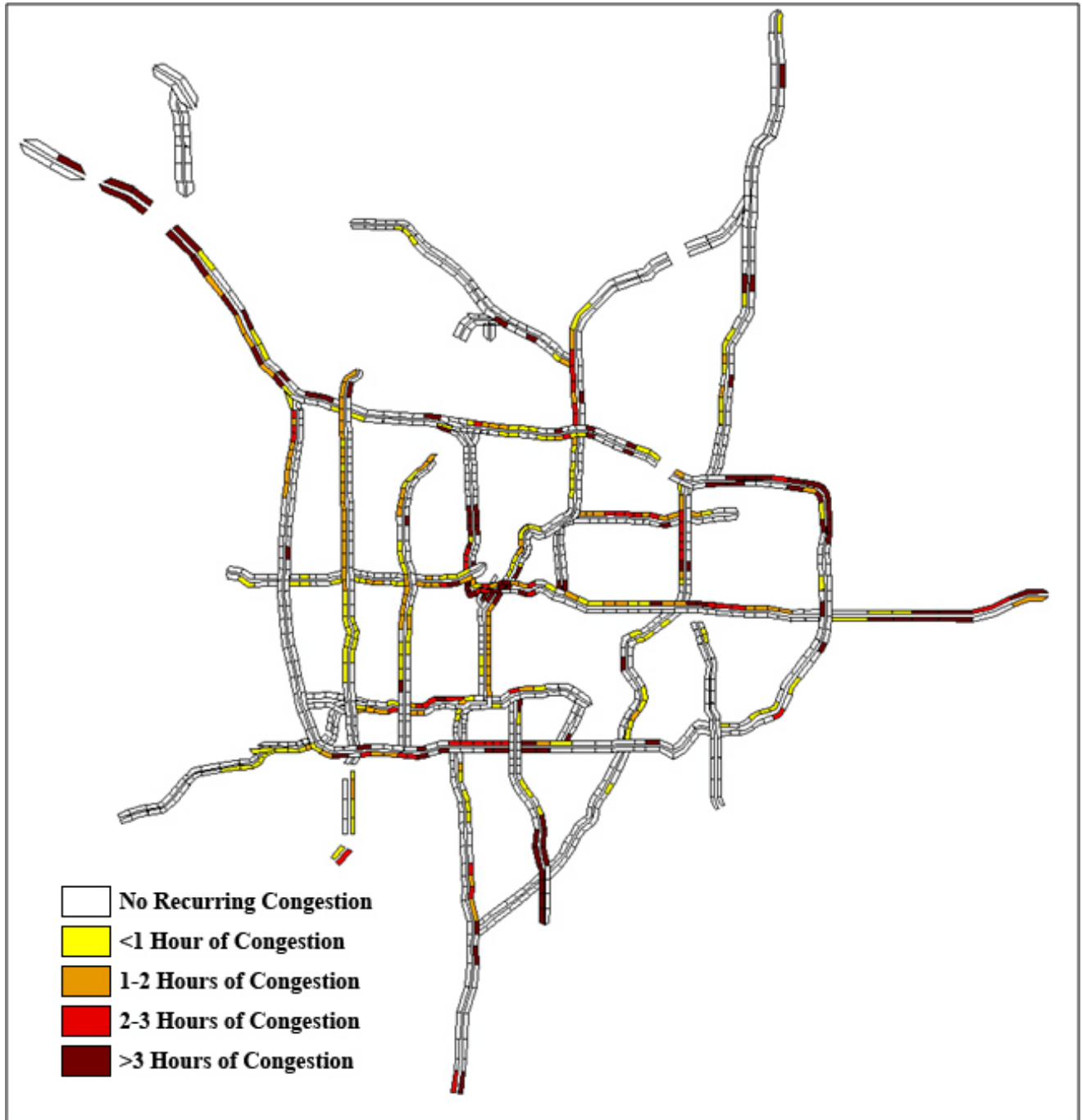


Figure 11 Speed-Based Congestion report visualization for AM-peak of October 2012 based on program cleaned data

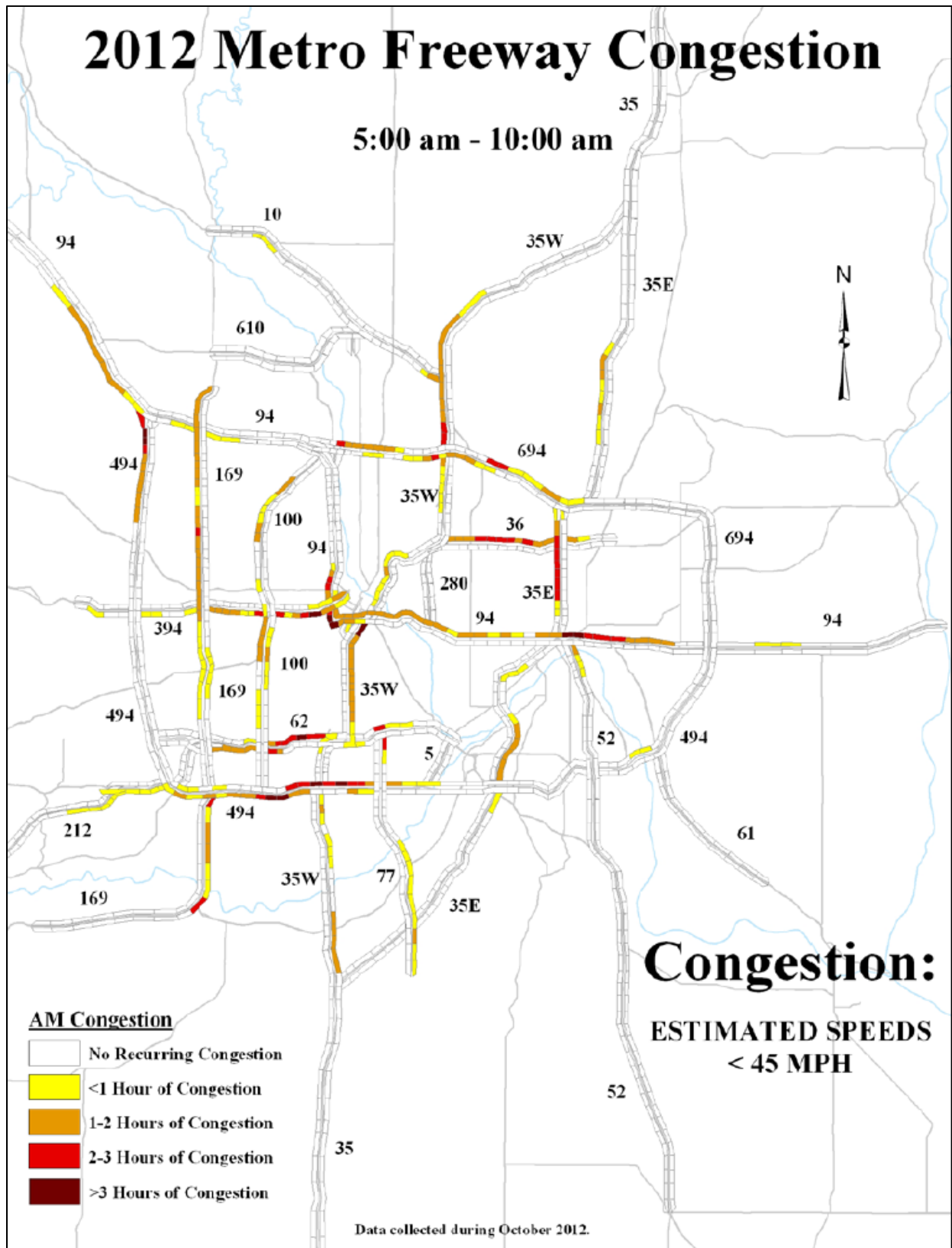


Figure 12 Speed-Based Congestion report visualization for AM-peak of October 2012 published by MnDOT

As can be seen in Figure 11 and Figure 12, there are some differences between the report generated by MnDOT and the one generated by the program. After examining the cleaned data produced by the program, several different reasons for these differences were found. The significant region of congestion along I-694 produced by the program is corroborated by the detector information found using DataPlot. Figure 13 below shows a sample of data for detector 6252. As can be seen, regular congestion during October would cause significant congestion during the 5-10 AM period. The other locations are similarly based on usable data.

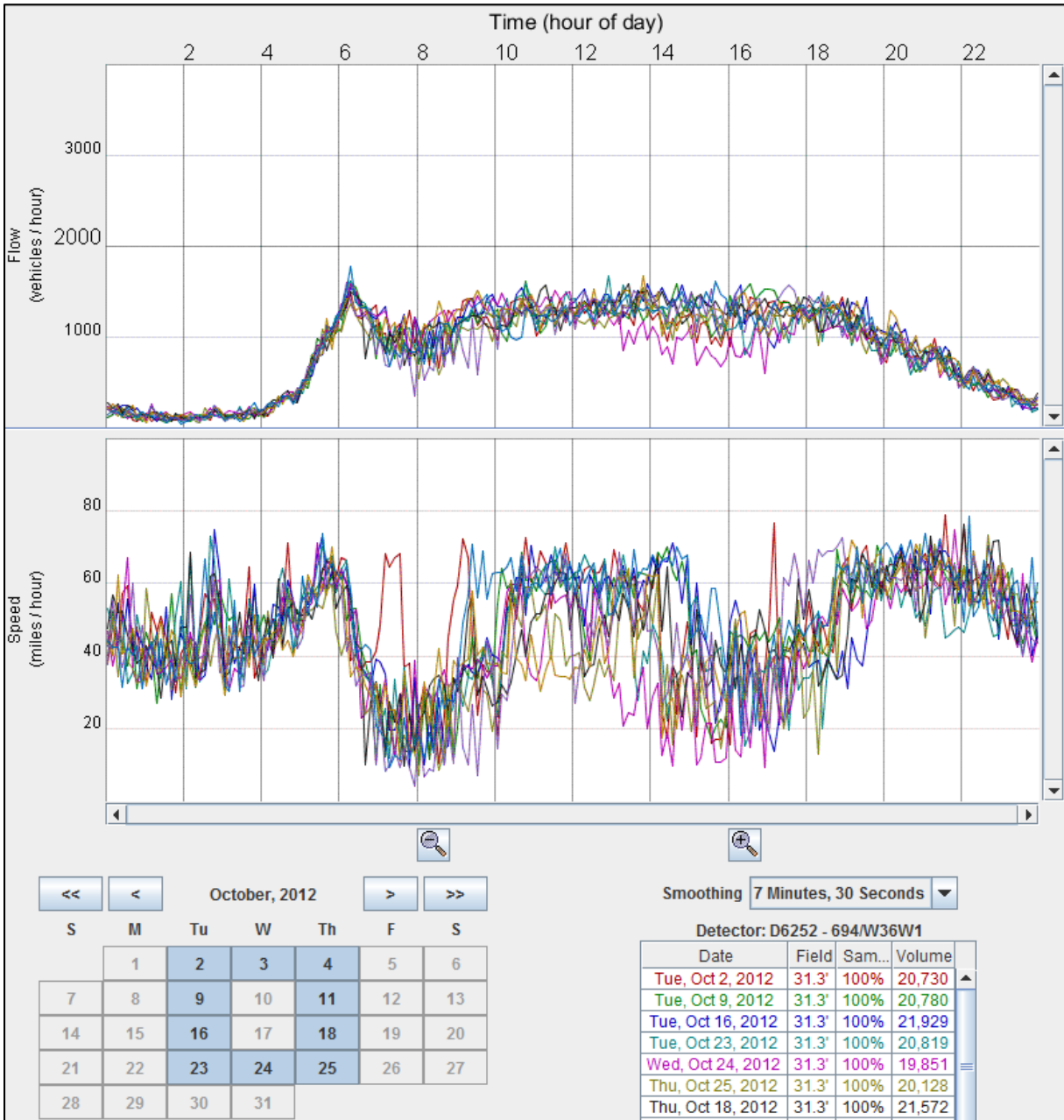


Figure 13 Sample data for detector 6252 along I-694, flow indicated in the top plot and speed in the bottom plot

In many of these cases, one of the detectors at a particular station is functioning normally (as above) while the remaining detector(s) are giving intermittent data or none at all. As a result, the single detector is used to generate data for the station and any existing congestion is noted. These locations may be construction areas or have some cause for the congestion that would not necessarily be useful to reflect in the annual congestion report. However, the program will not distinguish these locations. Any station with usable data will be analyzed and included in the report output.

Other locations showed poorly-estimated field length values that were skewing all speeds down such that a significant amount of uncongested activity is marked as congestion. Figure 14 below shows the speed and volume plots for detector 65 which is along I-94 just north of I-394. This location does not experience a breakdown, but as can be seen in the figure, the morning peak has a significant number of points that would fall below the 45 mph threshold for congestion.

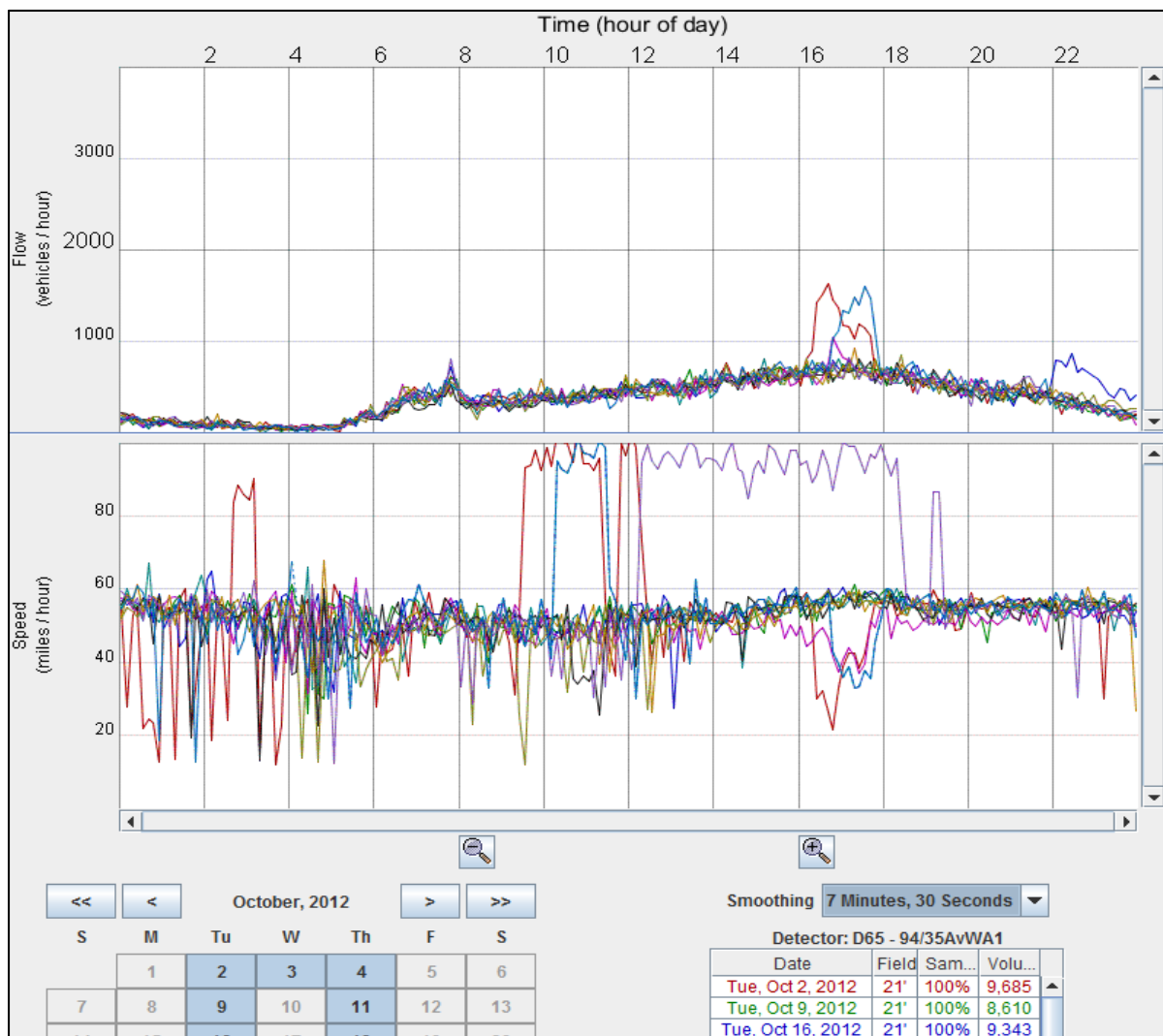


Figure 14 Sample data for detector 65 along I-94, flow indicated in the top plot and speed in the bottom plot

Possible Improvement

To get additional granularity with the SBC report, instead of tabulating the speeds only under the threshold of 45 mph, it is only incremental effort to additionally tabulate other thresholds. A series of thresholds (5-mph increments up to 45 mph) will allow for understanding not only where congestion is happening generally, but which locations are experiencing light, moderate, or severe congestion.

To improve the locations that have poor estimates of field length, there are several options. A standalone program or additional module could be created to examine a significant quantity of data for each detector, locate periods during the day with low density, and attempt to estimate the field length based on some assumed average speed (speed limit, speed limit + 2 mph, etc.). Alternatively, data available from TomTom GPS units could be used to estimate the field length in a similar fashion. These data, already in a database, would be queried to produce field lengths for any location and time segment.

In either of these above cases, estimating the field length based on data from overnight (which is the most predominant time of free-flow speed activity for many detectors) may result in a poor estimate as well. The composition of traffic during the late evening and overnight could differ significantly from the traffic composition during peak or midday hours. As a result, only midday uncongested times could be used to estimate the field length for these locations.

Based on the results obtained from October's AM peak, only a few locations would showed congestion due to poorly estimated field lengths. It may be possible to check these locations individually after the program produces cleaned data. The field length could then be modified and new speeds generated until the free-flow speed is correctly calibrated.

Congestion Intensity

This module builds on the existing structure of the SBC report, but accounts for the number of lanes present at each station. The methodology up to the tabulation of congested speeds is exactly the same as the SBC report. Once all the station totals are calculated, the total congested time for each location is simply multiplied by the number of lanes. Doing this accounts for the increased number of vehicles affected by congestion at wider multi-lane segments of the freeway network. Figure 15 shows a sample map for this report based on data from October 2013.

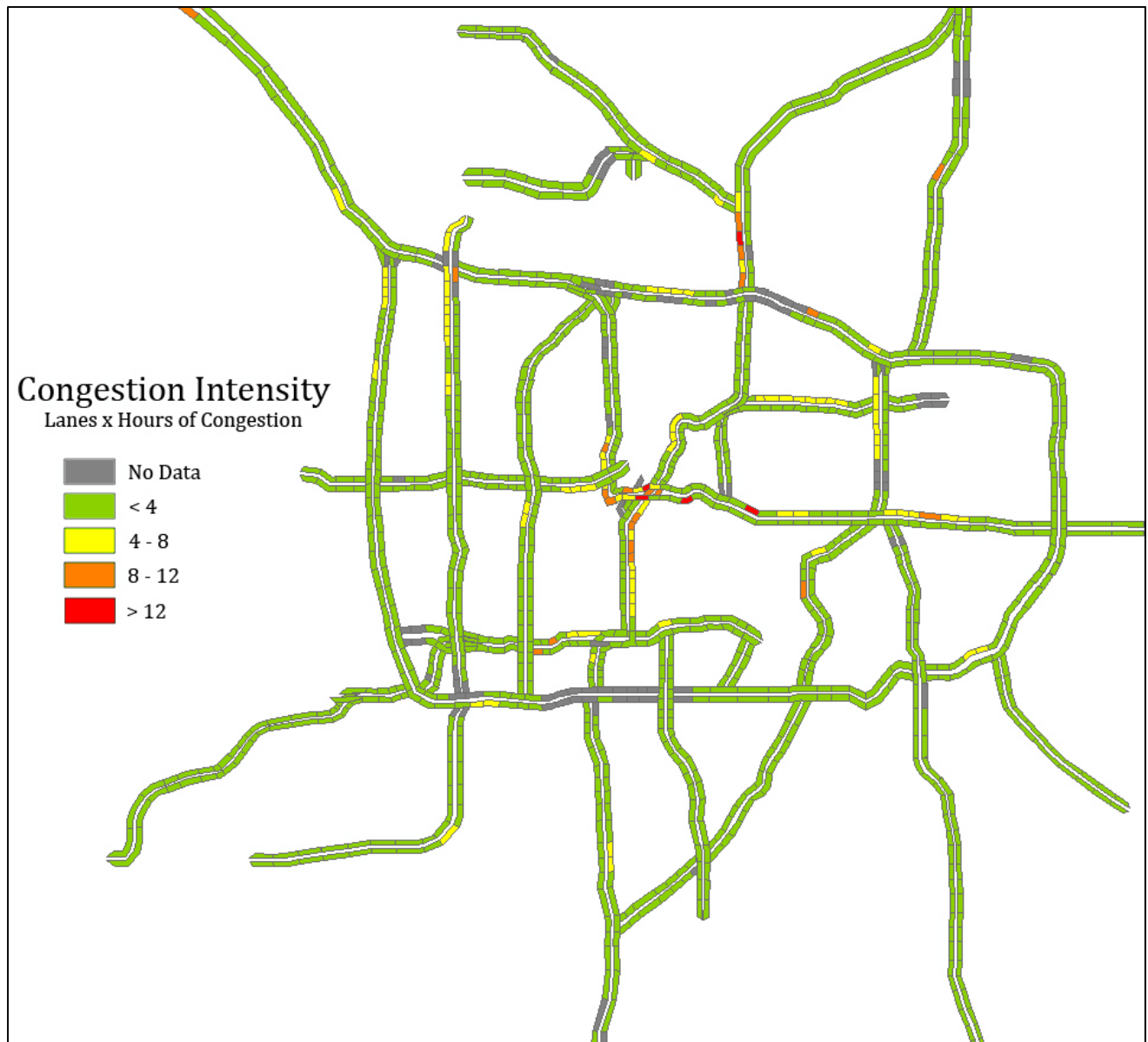


Figure 15 Congestion Intensity report for AM peak using October 2013 data

Lost Capacity Report Module

This module is concerned with estimating the severity of congestion at bottleneck locations by comparing flow and capacity. For each detector, two pieces of information are necessary for the module to operate: critical occupancy and capacity. Critical occupancy is the occupancy at which capacity is observed for the detector. These are stored as individual tables and can either be created using separate programs designed to estimate each for every detector in the network, or default parameters can be used for every detector in the network. The current module uses default values for all detectors.

With the capacity and critical occupancy parameters available, the Lost Capacity module follows the flow diagram shown in Figure 16. For every detector, the cleaned 30-second data produced by the program is aggregated to 5-minute data. Each 5-minute segment is compared against the critical occupancy to look for times of congested behavior. If congestion is located, the

difference between flow (5-minute volume) and capacity (adjusted to 5-minutes) is added to the lost capacity for this detector. The sum of lost capacity across detectors is used for the station lost capacity, and the median for each station across all selected days is used for the report. These station medians are used to find the average lost capacity per hour for each station (total lost capacity divided by period length). Figure 17 shows a sample of the output from this report.

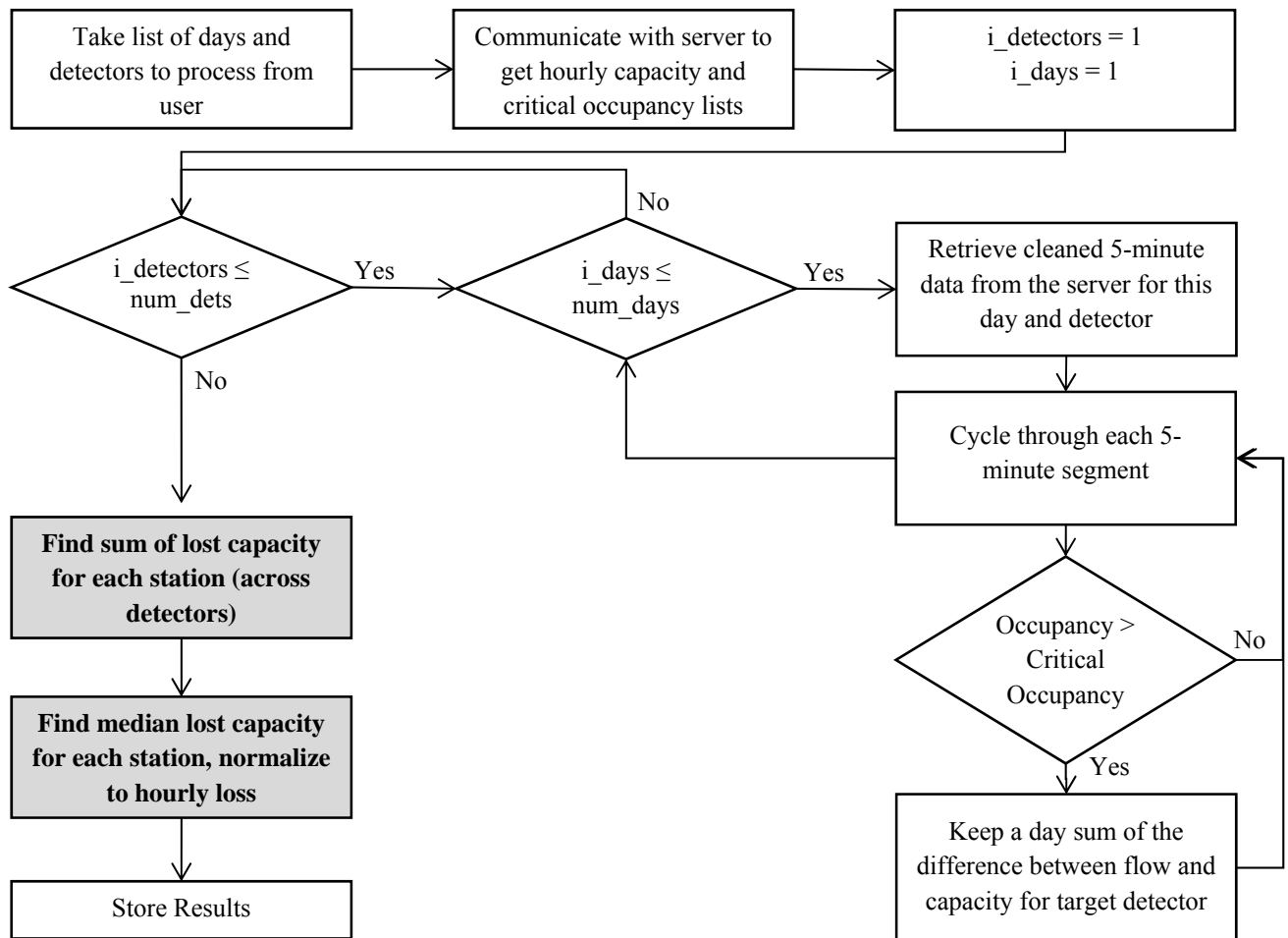


Figure 16 Lost Capacity module methodology flow diagram

2012 Metro Freeway Lost Capacity

5:00 am - 10:00 am

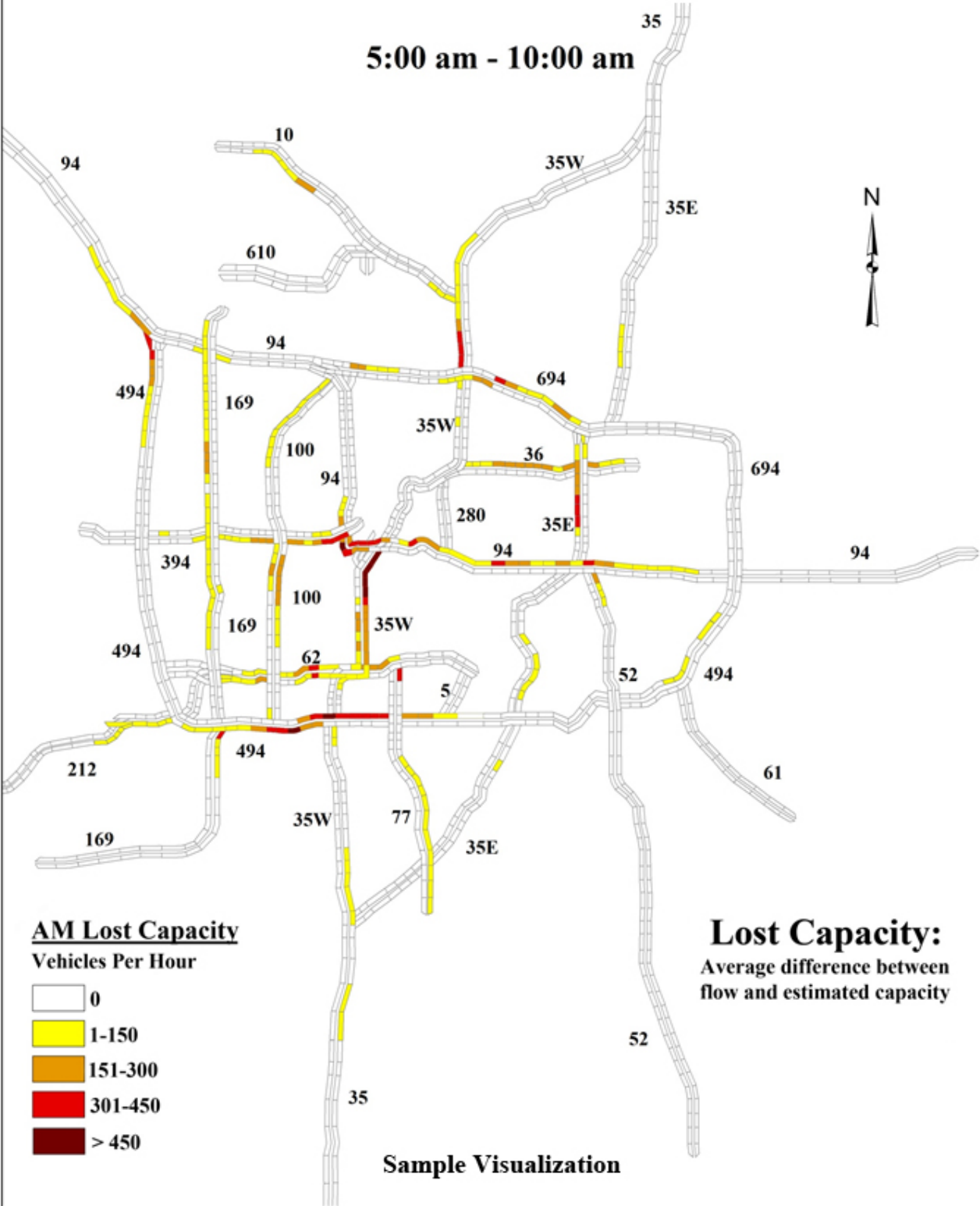


Figure 17 Lost Capacity report visualization

Possible Improvement

The original concept for the Lost Capacity report focused on identifying only bottleneck-related congestion. As congestion occurs across the network, vehicles approaching the breakdown must queue. As this queue grows, the congested region expands upstream of the bottleneck. The report concept involved identifying which stations within a congested region reported a traffic breakdown due to bottleneck effects at that station and which were congested merely due to queuing. With only those locations isolated, the report would measure the ‘lost potential’ at that bottleneck by comparing flow against capacity. (In fact, ‘lost capacity’ is likely not the best name for the report but was used consistently throughout the development process and was as such written into this version of HART.) By identifying the lost potential at each bottleneck, each location can be ranked by severity and management strategies and/or reconstruction investments can be targeted to give the greatest return.

In testing this original concept, the algorithm to identify the correct stations for the report had difficulty with stations downstream of the bottleneck which had low speed but were not congested. As a result, the check for queuing was removed and any location experiencing congestion will generate ‘lost capacity’. This does not change the ultimate goal of the report, but does leave some of the interpretation to the user in terms of identifying bottlenecks as opposed to queue backups.

Available Capacity Report Module

Related to the Lost Capacity module, the Available Capacity module is concerned with comparing flow and estimated capacity for detectors throughout the network. However, available capacity only occurs for locations that do not experience congestion during the time window selected (AM-Peak, PM-Peak, etc.). The purpose of measuring available capacity is to show how much additional ‘room’ each segment of the roadway has for additional use. As bottleneck areas are identified with the help of the Lost Capacity report, MnDOT planners can judge whether downstream locations will also need improvement as bottlenecks are alleviated.

Figure 18 shows the methodology for Available Capacity which is very similar to the Lost Capacity diagram. Cleaned 30-second data is aggregated to 5-minute values and checked against critical occupancy. If all 5-minute segments of a day are under the critical occupancy threshold, the 5-minute data is aggregated to 15-minutes. From these, the 15-minute segment with the highest volume from the day is selected and compared against the capacity for the detector. The difference is stored as the available capacity for that detector-day. The sum of available capacity across all detectors in a station is saved as the station’s available capacity, and finally the median available capacity is determined for each station across all selected days. Figure 19 shows a sample visualization of this report.

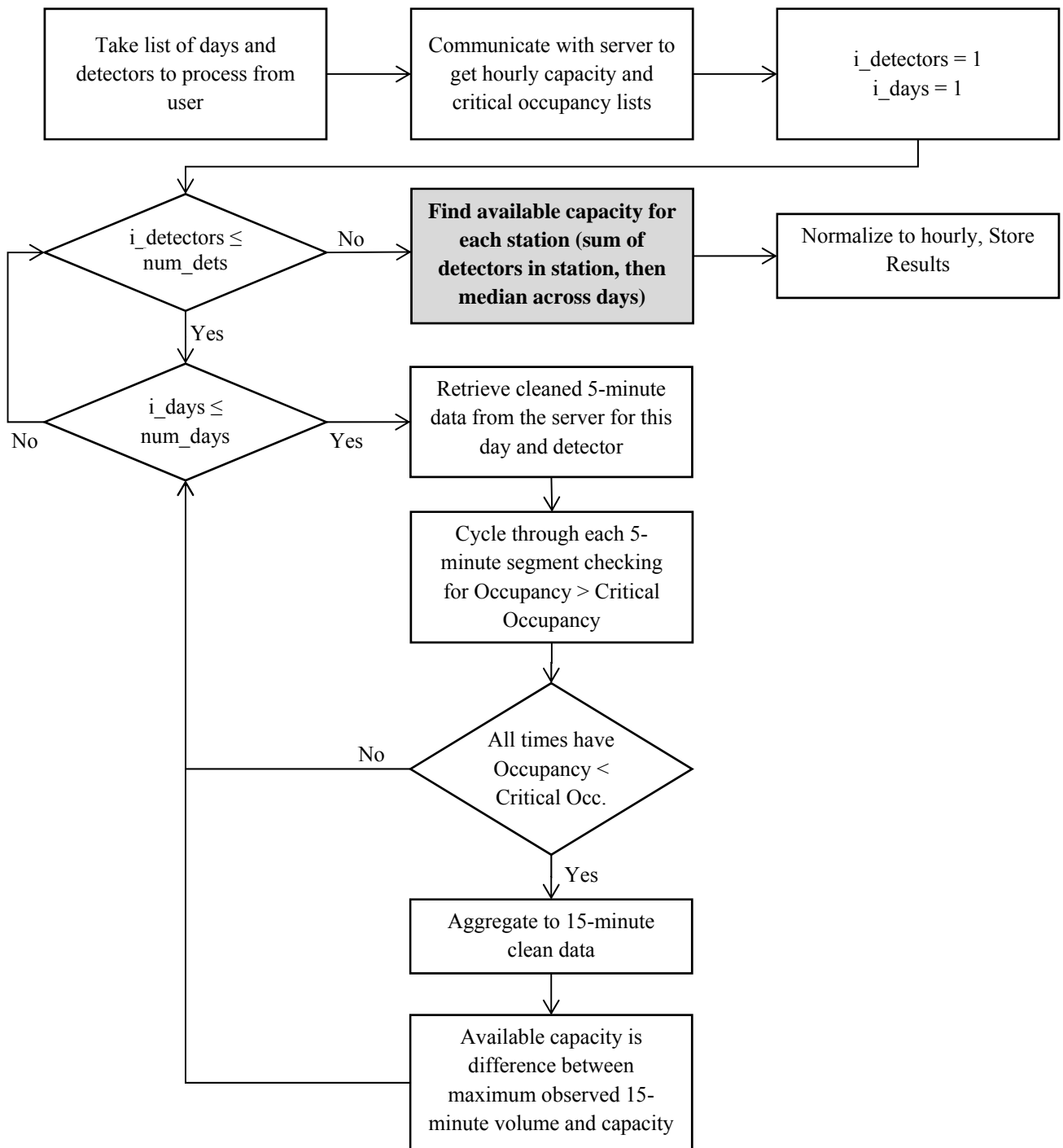


Figure 18 Available Capacity module methodology flow diagram

2012 Metro Freeway Available Capacity

5:00 am - 10:00 am

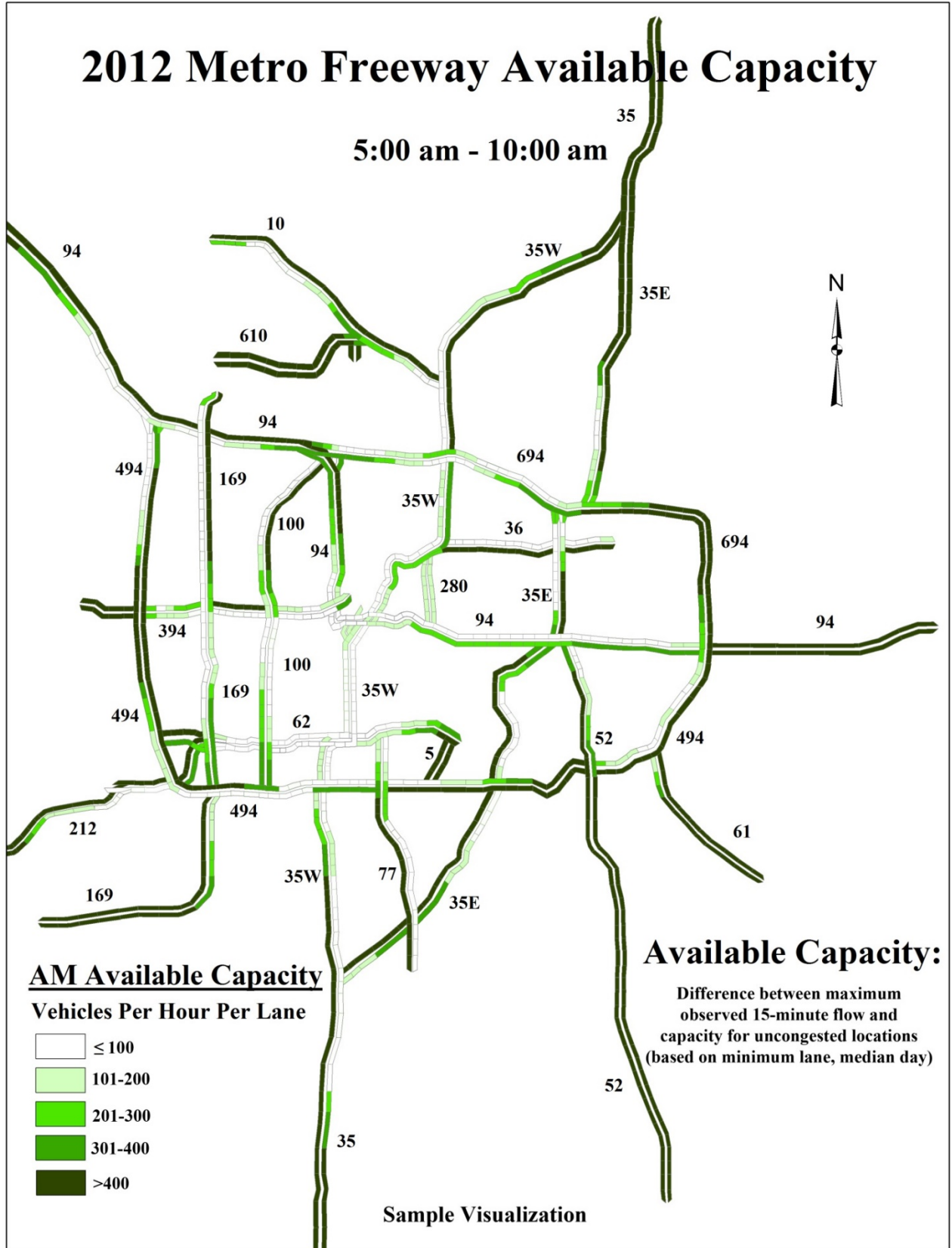


Figure 19 Available Capacity report visualization

Maximum Throughput and Total Throughput

These two modules are closely related and should be considered together. Figure 20 shows the flow diagram for the two modules together. For each detector, the 30-second cleaned data from the program is aggregated to 15-minute data to improve the stability of flow tabulations. For each 15-minute segment, the detectors in each station are summed to create 15-minute station data. For each day, the maximum 15-minute volume is stored along with the total volume observed for the station across the selected period. To summarize across all days in the selection, the median values for each maximum and total throughput are calculated and saved. Figure 21 and Figure 22 show the visualizations for these two reports. Note that these values are for the user-selected time segment and total throughputs will therefore vary widely between shorter and longer time windows. Care and discretion must be used when interpreting the results of this report. It may be useful to compare non-similar segments (such as AM v. PM peak periods) for total throughput, but such segments should not necessarily be treated as equivalent.

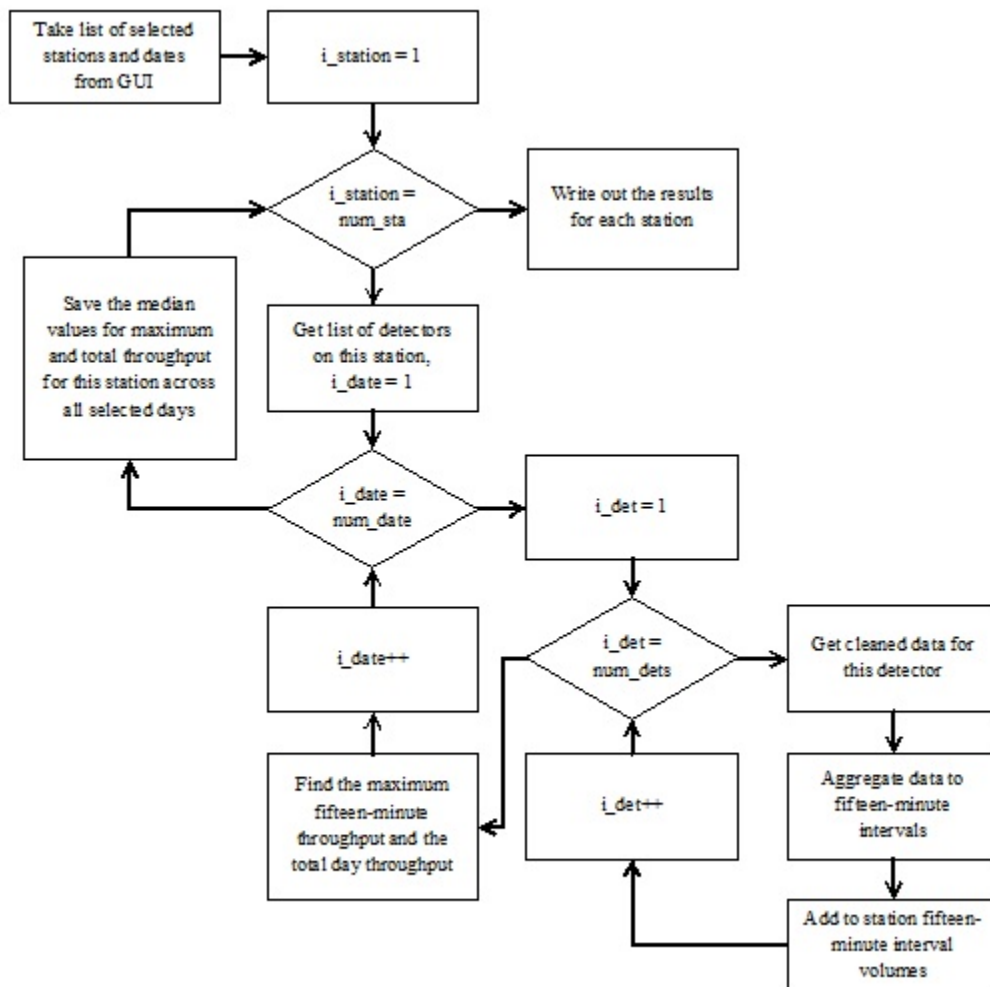


Figure 20 Maximum and Total Throughput modules methodology flow diagram

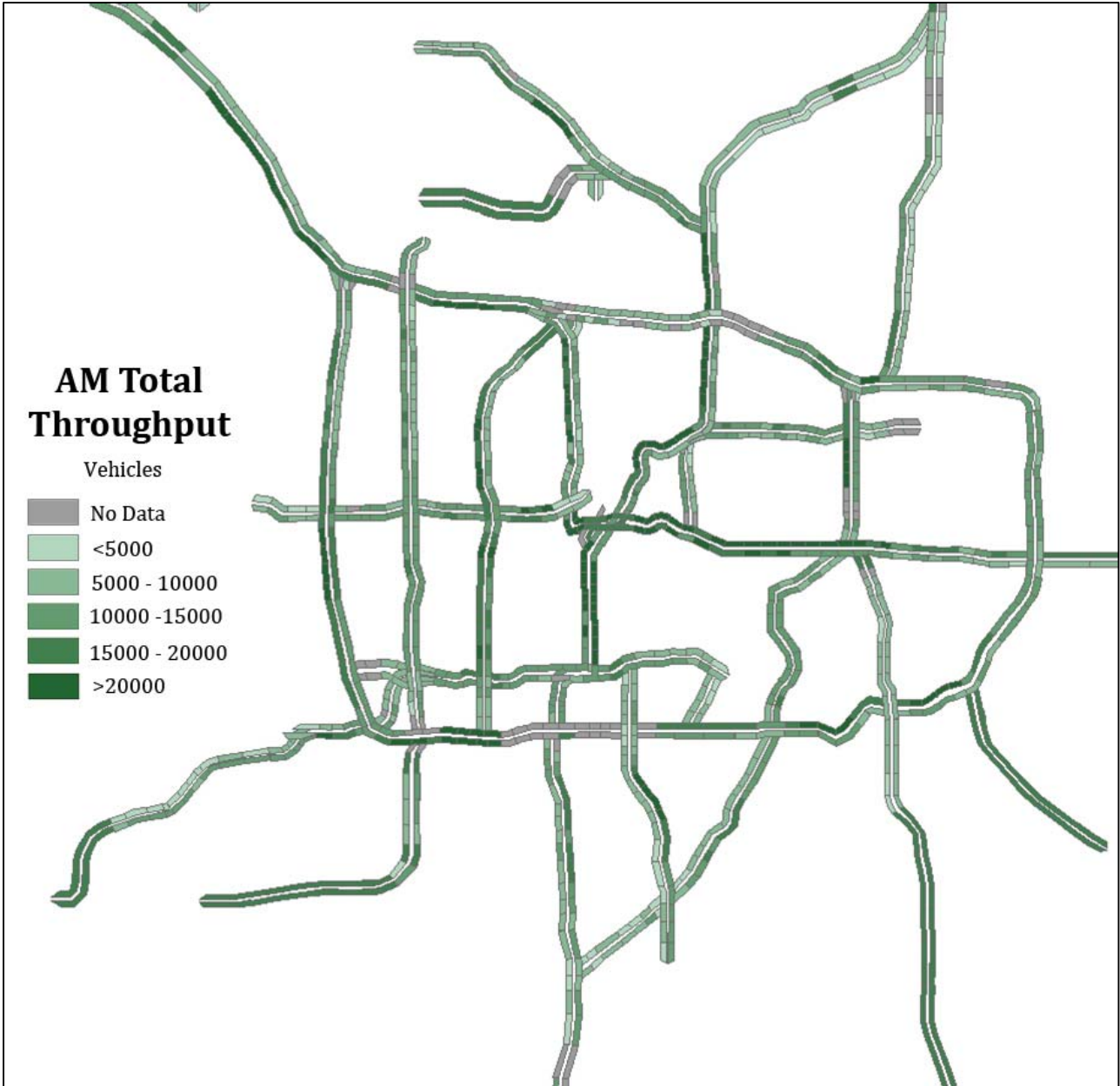


Figure 21 Total Throughput report for AM peak using October 2013 data

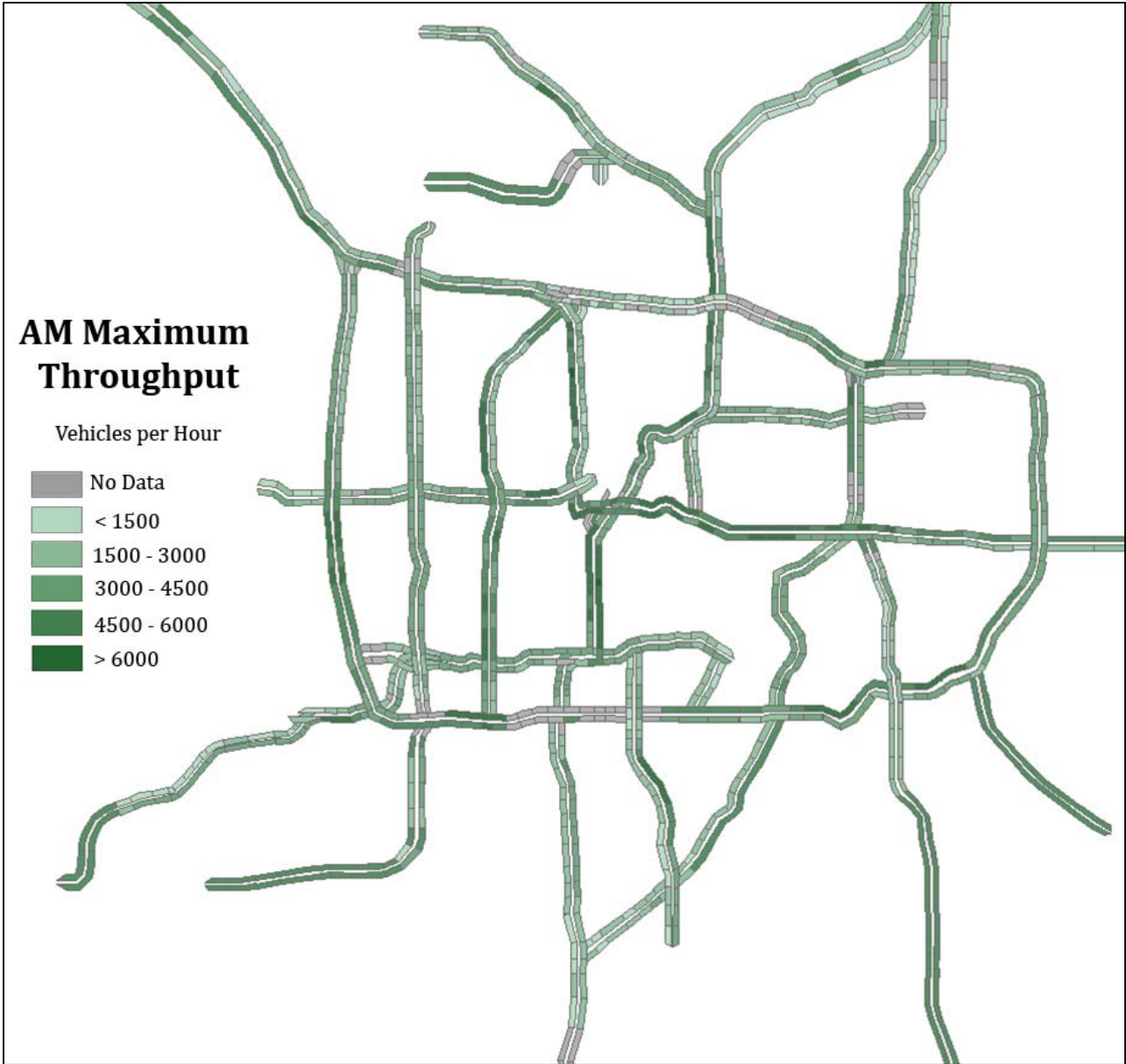


Figure 22 Max Throughput report for AM peak using October 2013 data

Report Visualization

The Performance Measure Visualization module updates a database file tied to an RTMC report generation shapefile with data from each of the report modules. Once a database file is selected (at the start of report generation), the results for each selected report module are generated, sent to the PMV, and added to the dbf. Each value is keyed by station and placed into an appropriately named field (based on the time window selected in the main user interface and which report is being generated). Figure 23 below shows a flow diagram of the PMV module.

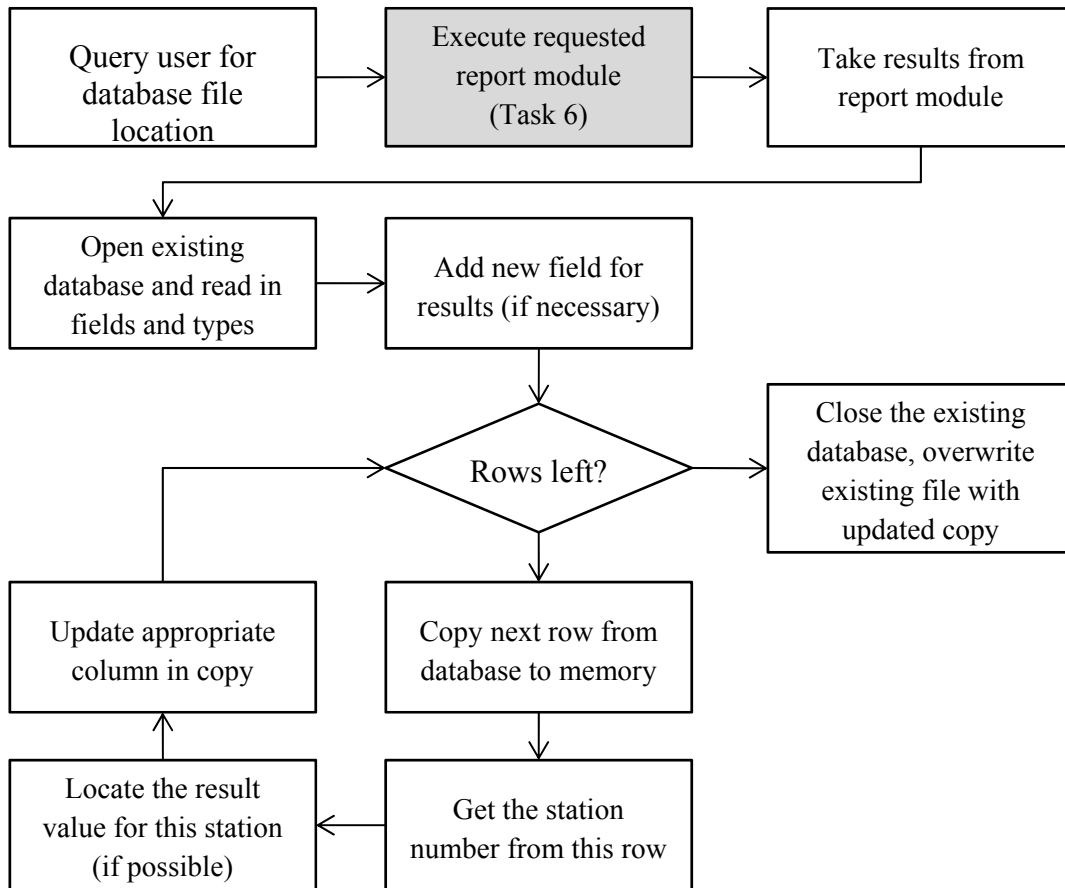


Figure 23 Flow diagram of Performance Measure Visualization module

If a station is within the existing RTMC database file but *not* in the requested report from the tool the value will be updated to a special value (-1 is currently used for each report) to indicate that the station is not included. Figure 24 shows a sample of a report generated by the module for Highway 77 northbound. Since no other corridor was selected for this report, all others are marked as 'Not Included' while each station in Highway 77 northbound shows the appropriate value.

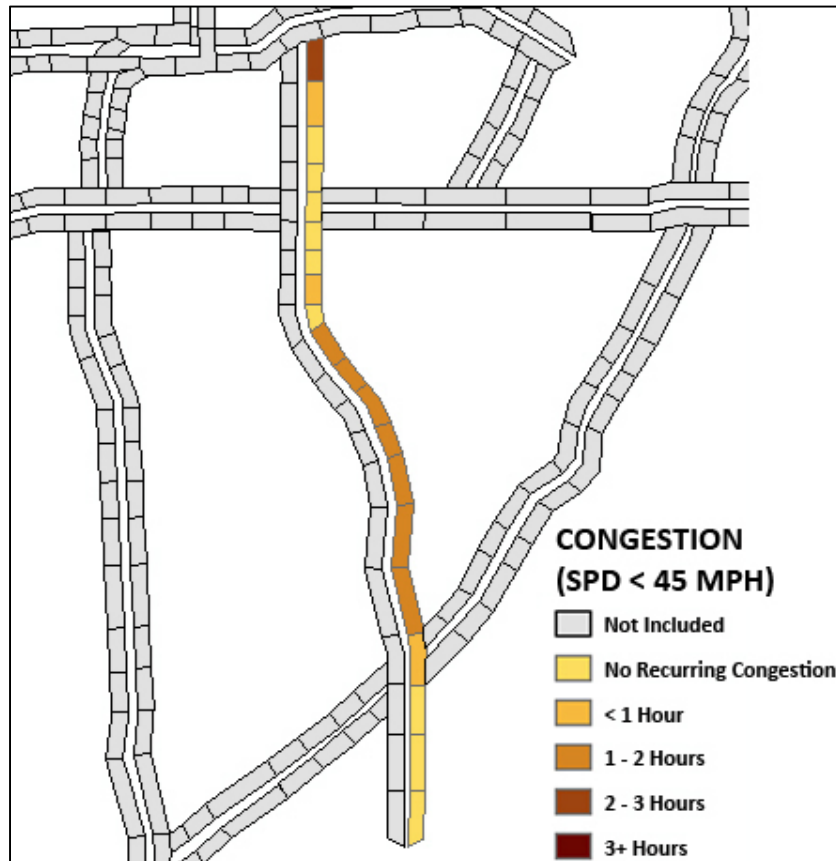


Figure 24 Sample output from Speed Based Congestion Report

5. Discussion

HART was designed primarily to automate and simplify the generation of the RTMC's annual Congestion Report. As was noted in the section detailing that report, additional granularity could be added to describe the type of congestion occurring across the network. The other report methodologies similarly are open to modification and improvement. In particular, 'lost capacity' and 'available capacity' were defined through significant discussion as HART was being developed.

The current lost capacity report examines all detectors across the network and counts any capacity drops due to congestion. An alternative (possibly a different, supplementary report) was proposed which involved isolating only the stations which are immediately at the bottleneck point for each congested area. This would highlight the bottleneck locations across the network and give some indication of the severity of the bottleneck through the lost capacity measure.

Lost capacity is also measured across the entire period of congestion instead of isolating the 'worst' moment of congestion. Available capacity is currently defined as the difference between capacity and the maximum flow experienced. A 'total available capacity' measure could be created which finds the total difference between flow and capacity across an entire period.

Maximum and total throughput, while relatively simple measures, are also open to improvement. For each of these, instead of a single value being given as an average across a number of days or weeks, additional parameters could be calculated to capture the variation in total and maximum throughput observed across the network.

Finally, HART relies on several sets of constants which, for this version, are predefined for each detector. However, additional modules (or standalone programs) could be developed to improve these constants: field length, capacity, and critical occupancy. Field length, although calibrated for each detector, is kept as a single value which does not account for the natural variation in the composition of traffic each day (or even across the day). Capacity and critical occupancy are standard values which are not calibrated to each detector. Automatically estimating and updating these values could significantly improve the results generated by HART.

6. Conclusion

The HART program developed in this report completes the “Expanding and streamlining of RTMC freeway network performance reporting methodologies and tools” project. Through its user interface, HART allows for significant flexibility in isolating corridors, dates, and times on which to generate report information. HART automatically identifies and corrects for several forms of error within the loop-detector data collected across the network. Small-scale errors are corrected in place to preserve as much existing data as possible. Large-scale errors are imputed wherever historical data is available to offer insight into the relationship between a broken detector and a nearby detector. Finally, a series of reports examine the cleaned data to create performance measure information with which to judge the behavior of the network and the effectiveness of implemented management decisions.

This is the first version of HART and only a first step toward streamlining and improving the performance-based management tools available for RTMC to monitor the health of the Twin Cities network. Through HART, the data available from the loop detectors across the network can be cleaned and stored for quicker and easier analysis than the previous methodologies. Further, additional reporting modules have been incorporated to widen the spectrum of performance measures available with which to judge the needs of the freeway system. HART offers RTMC the ability to more frequently assess the behavior of the network with a more consistent methodology and ultimately have more information with which to make planning and management decisions.

HART was developed using a similar structure and within the same coding language as existing tools at RTMC. This was done purposely to simplify the maintenance and use of the existing program, but also to allow MnDOT engineers to have the ability to implement changes to HART over time. As HART is distributed and used, modifications will likely be desired to further improve the tool. By developing in an environment that RTMC is already familiar with, HART can be a living program and grow as monitoring needs shift and change.

References

Minnesota Department of Transportation. *Metropolitan Freeway System 2012 Congestion Report*. Retrieved from <http://www.dot.state.mn.us/rtmc/reports/CongestionReport2012.pdf>

Appendix A – Installation Instructions

This installation package should include the following files (plain text) and folders (italicized):

- HART.jar
- HART.conf
- *HART_lib*
 - o Commons-logging-1.1.1.jar
 - o Commons-math3-3.0.jar
 - o Javadb-0.4.0.jar
 - o JRI.jar
 - o JRIEngine.jar
 - o PostgreSQL-9.2-1000.jdbc4.jar
 - o REngine.jar
- *R_modules*
 - o correctedOutlierSeries.R
 - o correctedOutlierSeries_orig.R
 - o crossCorrelation.R
- *GIS_and_Config*
 - o gpoly.dbf, gpoly.shp, etc.
 - o metro_config.xml

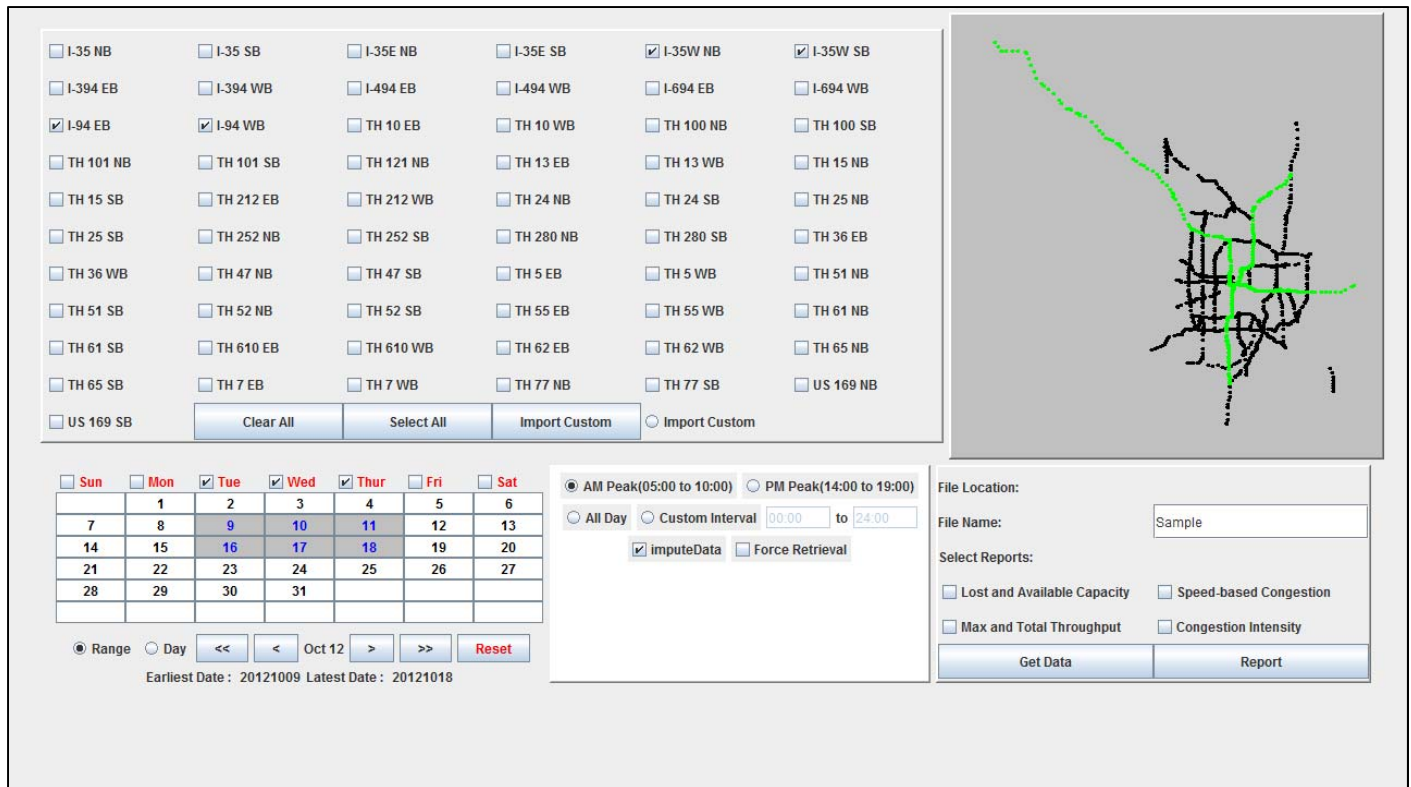
Installation Steps

1. Visit Oracle's website to download and install a current version of the Java JRE.
(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
On the page listed above, find the "Download" button just under the JRE header part way down the page. Accept the license agreement and select the appropriate installation exe (x86 for 32-bit machines, x64 for 64-bit machines). The default installation of the Java JRE is sufficient for HART.
2. Visit the R Project website to download and install a current version of R.
(<http://cran.r-project.org/bin/windows/base/>)
On the page listed, begin the download for Windows at the top of the screen. The default installation of R is sufficient for HART.
3. After R is installed, open R by right clicking and selecting Run as administrator (you must select "R x64" for 64-bit Windows and "R i386" for 32-bit Windows):
install.packages("rJava")
 - a. If this command is successful, a window will pop up asking for a CRAN mirror. Select a nearby mirror and select ok. A short message should be printed when the installation is successful. R can be closed.
 - b. If a window appears asking to "Use a Personal Library", then R was not opened with sufficient privileges. Try step 3 again, making sure to Run as administrator.
4. In Windows System Properties (Control Panel -> System -> Advanced system settings) select Environment Variables. A User variable named PATH should be added or modified to include "C:\Program Files\R\R-3.0.1\library\rJava\jri\x64;C:\Program Files\R\R-3.0.1\bin\x64;"
 - a. These are the default installation locations for these two folders, but they may differ system to system. Note that these two example paths are for a system running 64-bit Windows. For a 32-bit machine, the "x64"s should be replaced by "i386"s.

- b. After this variable is created or modified, select Ok and close the popup windows. It is recommended that the user be logged off and logged on so that the system variables are updated.
5. Open the HART.conf file included with this package with a notepad editor. On each line, update the information on the right (**do not** change the #variable names).
 - a. #databasename should be the name of the database to be used for data storage as defined within the PostgreSQL manager. To set up a new PostgreSQL server to handle the data produced by HART, see the included New PostgreSQL Server instructions.
 - b. #databaseurl is the IP address and port for the above named database. By default, PostgreSQL uses port 5432.
 - c. #databaseuser is the username for accessing the PostgreSQL server.
 - d. #databasepass is the password for accessing the PostgreSQL server.
 - e. #rworkspace is the path to the included folder called “R_modules” and **must use forward slashes** (“/”) as directory separators instead of backward slashes (“\”).
 - f. After all values are updated, save and close HART.conf
6. Installation of HART is complete and HART.jar may now be double-clicked to start the program.

Appendix B – User Manual

The HART user interface (see figure) provides an intuitive layout and features that allow for customization of the input and output data. The various features included within the user interface are described with some instruction below. Note that the first step of running the HART program is selecting a configuration file (either a tms_config.xml or metro_config.xml file) which describes the Twin Cities network. This is the same file that is used for IRIS.



Highway Selection

All interstate, U.S., and trunk highways within the Twin Cities metropolitan region are included as selections within the user interface with each highway generally represented twice, once per applicable direction. When selected, the applicable highway(s) become highlighted in green on the indicator map to the right. For convenience, “Clear All” and “Select All” options have been included.

As of this version, the presence of this specific grouping of highways within the user interface is hardcoded in, meaning they will be present regardless of the information provided in the configuration file selected upon program startup. Note that if a configuration file does not include data for a particular highway, HART will be unable to produce any data or reports applicable to the highway regardless of whether it is selected in the user interface.

To import a custom file, change the “Import Custom” toggle option to be selected, and press the “Import Custom” button to the left. You will be prompted to select the file containing the custom list of detectors and stations of interest. This file must be of the .txt file type, and must be

formatted so that each line contains only the name of one detector, indicated in the .txt document with a “d”.

Time and Date Selection

An interactive calendar is provided to allow for easy selection of dates of interest. Each day of the week is designated with a checkbox that dictates whether the applicable day may be selected across the range of chosen days. The “Range” setting allows for the first and last days in a range to be clicked sequentially, and all the applicable days between them are automatically selected. Alternatively, the “Day” setting requires each day to be manually clicked in order for it to be selected. The “Reset” option clears all selected days. When there are actively selected dates, text below the calendar shows the earliest and latest dates chosen in order to assist in keeping track of selections.

Four time options are available: AM peak, PM peak, all day, and custom interval. AM peak is set from 05:00-10:00, while PM peak is 14:00-19:00. “All Day” includes the 24 day between midnights. Custom intervals may be precisely set to the minute.

Note that HART will attempt to gather data for the requested dates and times even if the detectors within the given configuration file have no data for that window. Some care should be used in selecting an appropriate configuration file to most closely match the state of the network as the requested time.

Two additional options are shown below the time selection area. The “Impute Data” option allows HART to use the imputation module, while unchecking the option will force HART to refrain from fixing data related to broken detectors. The “Force Retrieval” option ensures that HART retrieves data from the RTMC server regardless of whether the corrected data server contains data for the detector(s) in question.

Output Options

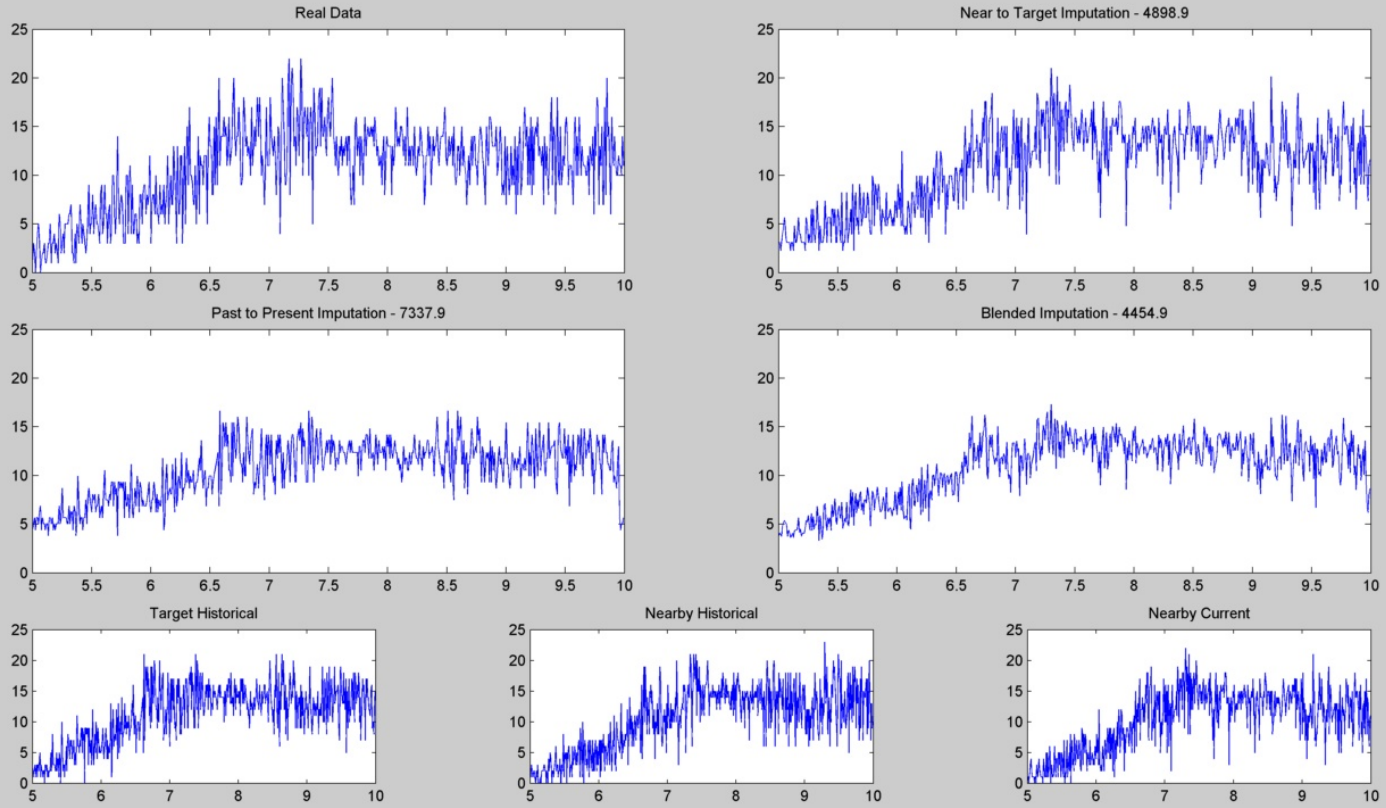
The four reports included in the program are available with check box icons in the lower right. The output options are either “Get Data” or “Report”. These buttons are meant to be used in sequence: after selecting corridors, dates, and a time, data must first be retrieved and cleaned using “Get Data”. Once that portion of the program completes, reports may be selected and generated using “Report”.

The intermediate modules of the program which perform retrieval, cleaning, and imputation now run without manual intervention in the background after “Get Data” is selected. The final modules which generate the reports and apply the results to the existing RTMC report structures require only one point of manual intervention. Immediately after selecting “Report” the user is asked for a database file (.dbf) to modify. This database file is the one associated with the ArcGIS shapefile which is used for generating the RTMC reports. Once specified, the reporting modules automatically update the appropriate values within the database file. By reloading the ArcGIS report, the updates made by the program are automatically present.

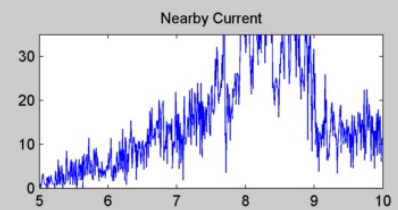
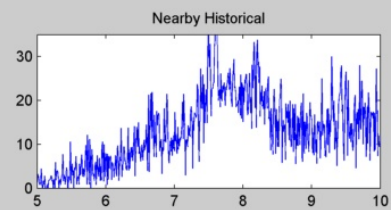
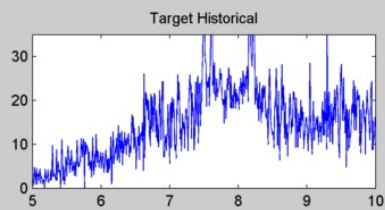
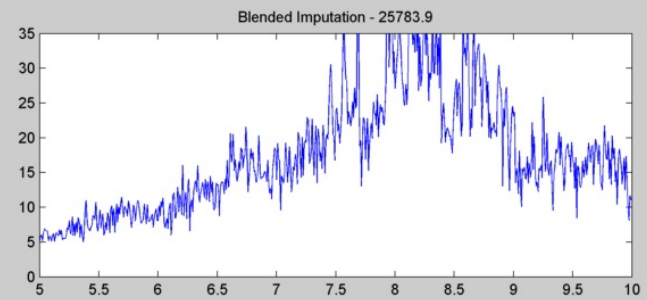
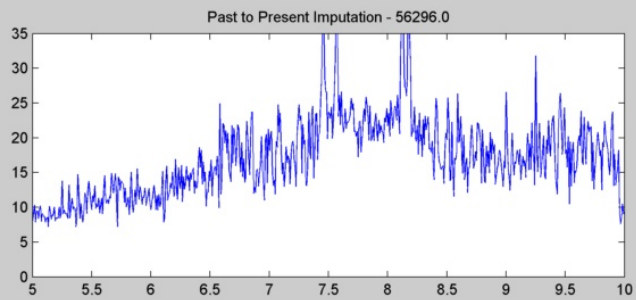
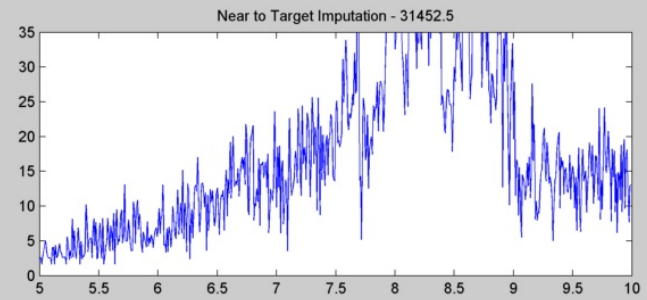
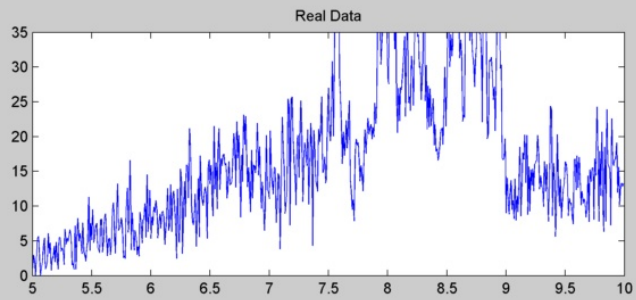
Appendix C – Imputation Figures

#	Target	Nearby	Current Date	Historical Date
1	339	340	10/5/2010	10/21/2010

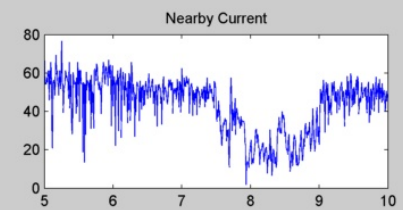
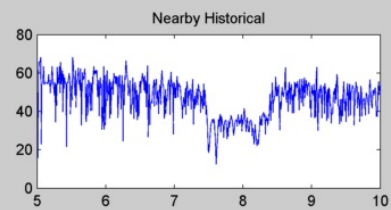
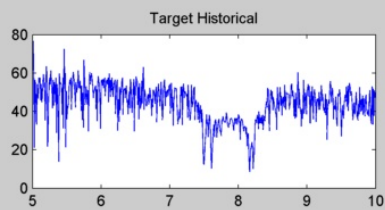
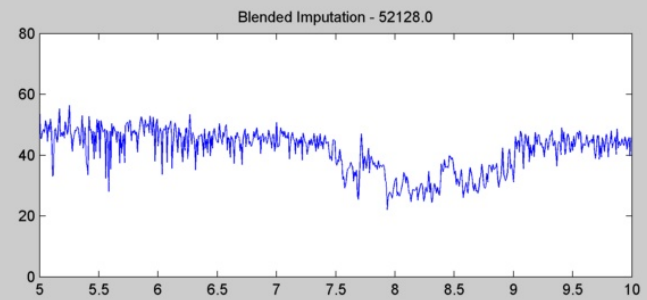
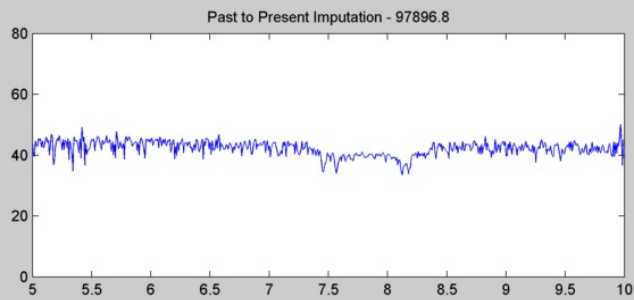
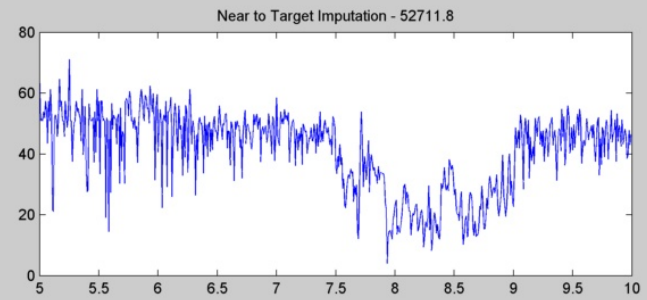
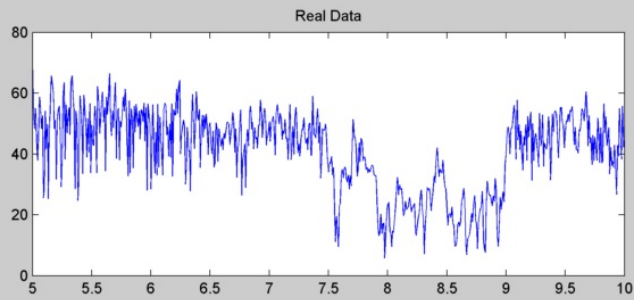
Volume - Correlation: 0.802



Occupancy - Correlation: 0.802

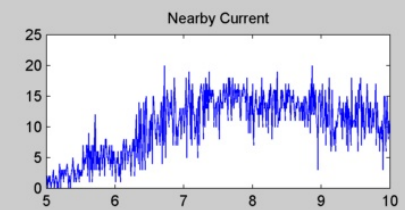
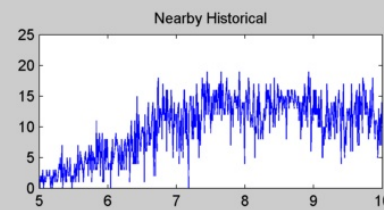
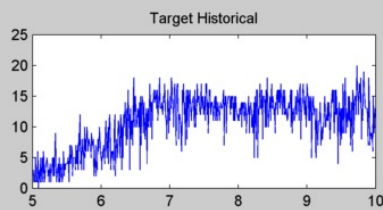
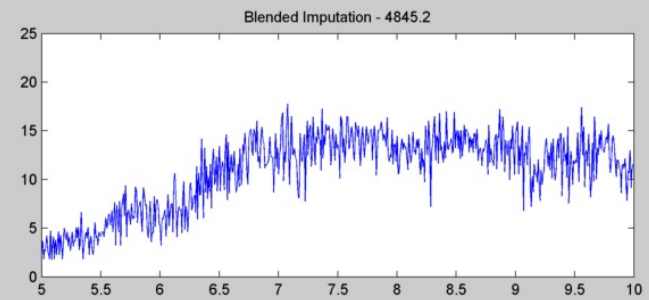
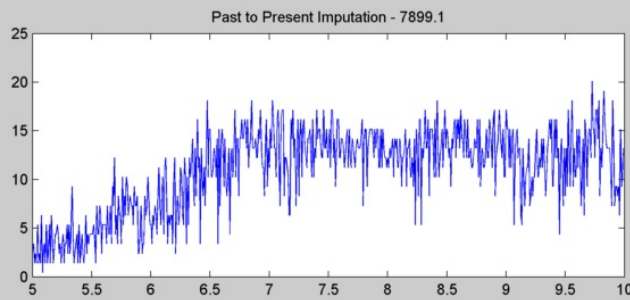
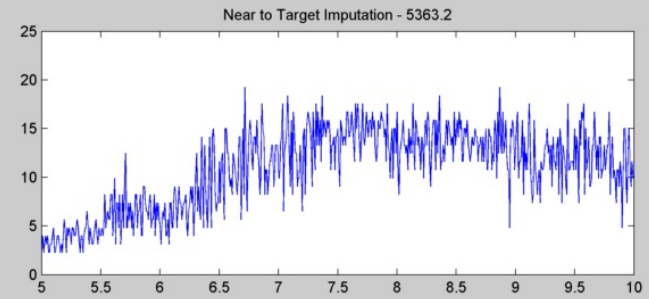
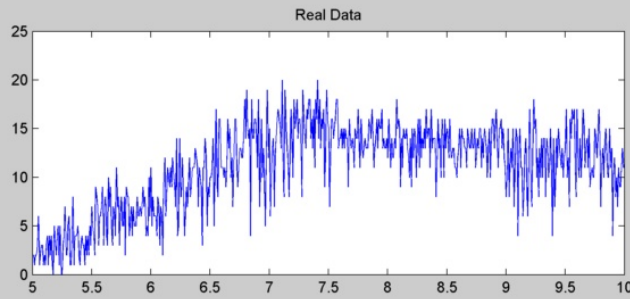


Speed - Correlation: 0.802

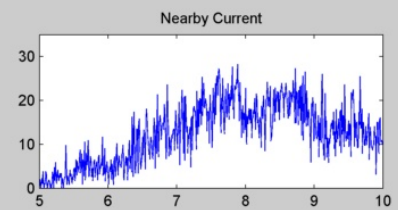
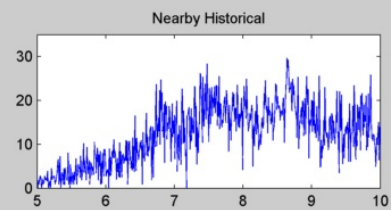
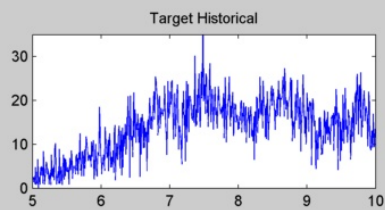
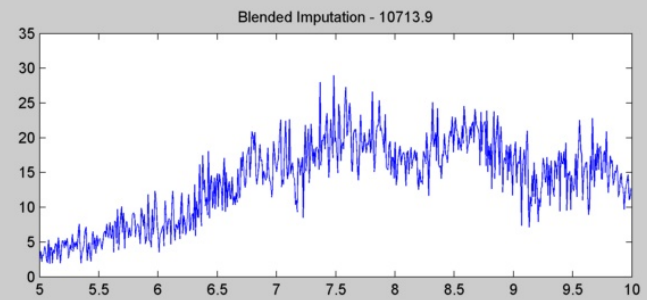
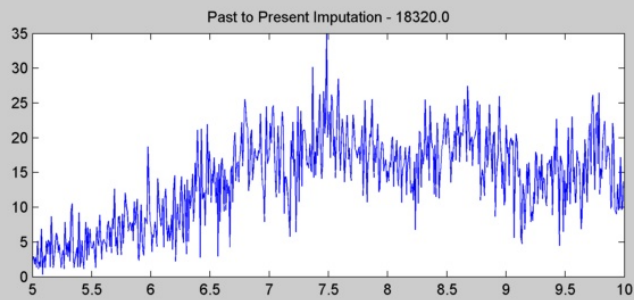
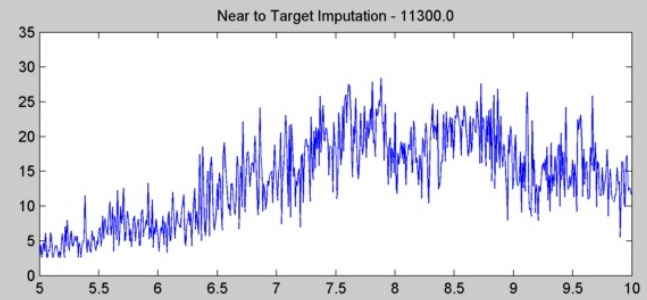
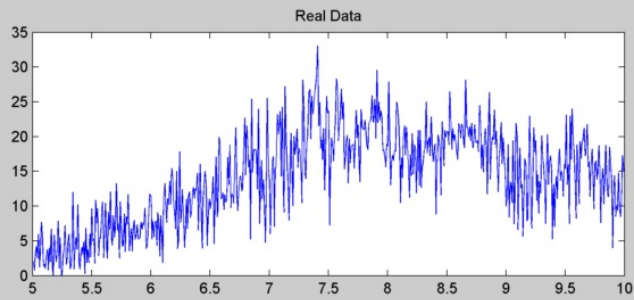


#	Target	Nearby	Current Date	Historical Date
2	339	340	10/26/2010	10/27/2010

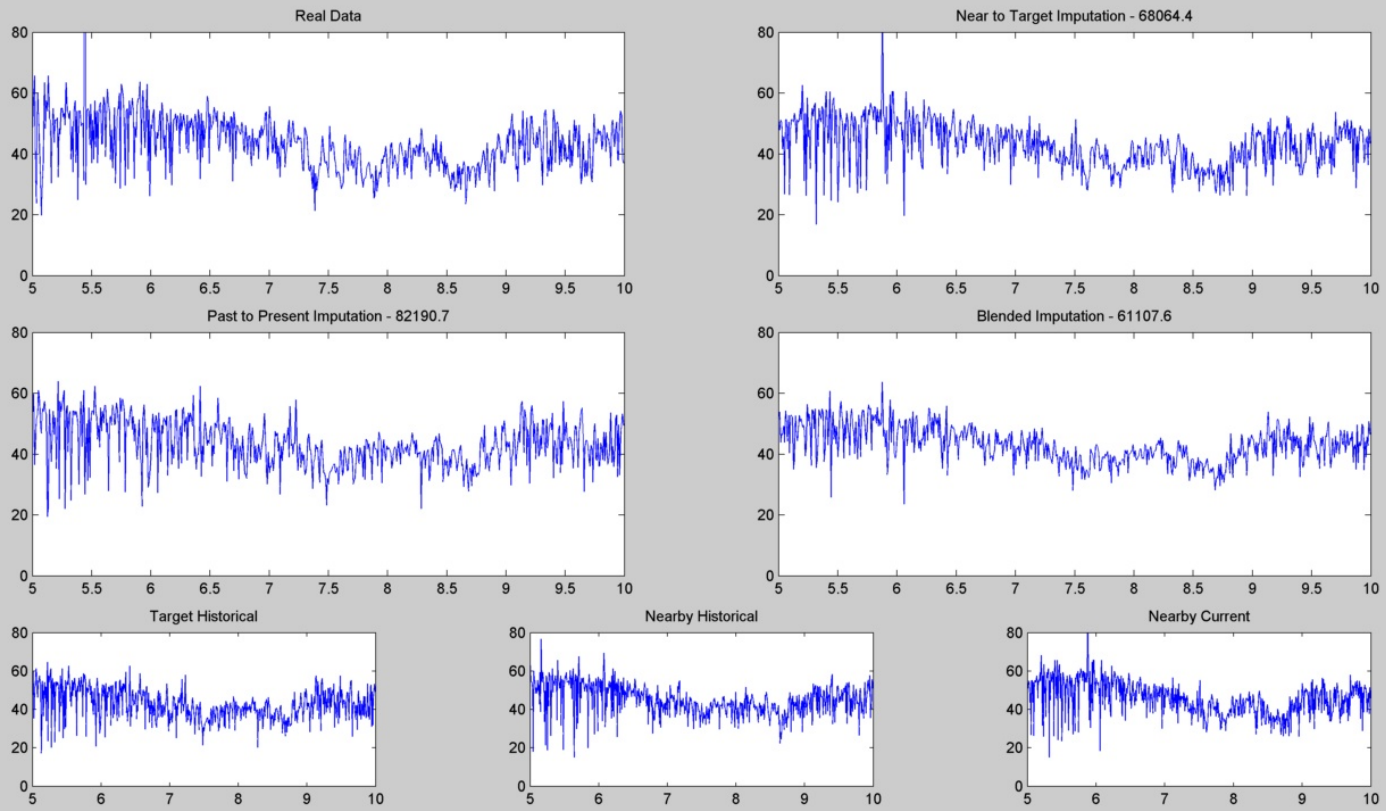
Volume - Correlation: 0.941



Occupancy - Correlation: 0.941

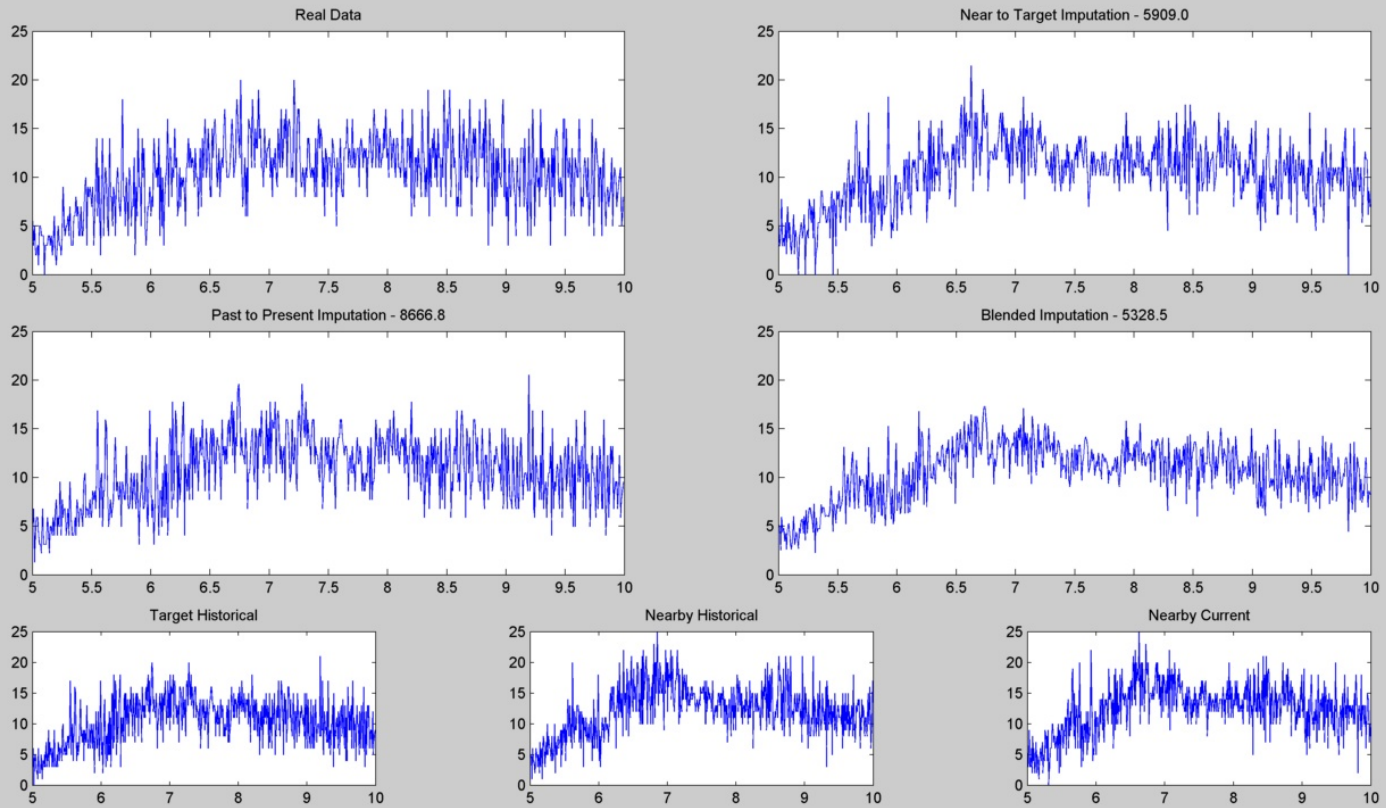


Speed - Correlation: 0.941

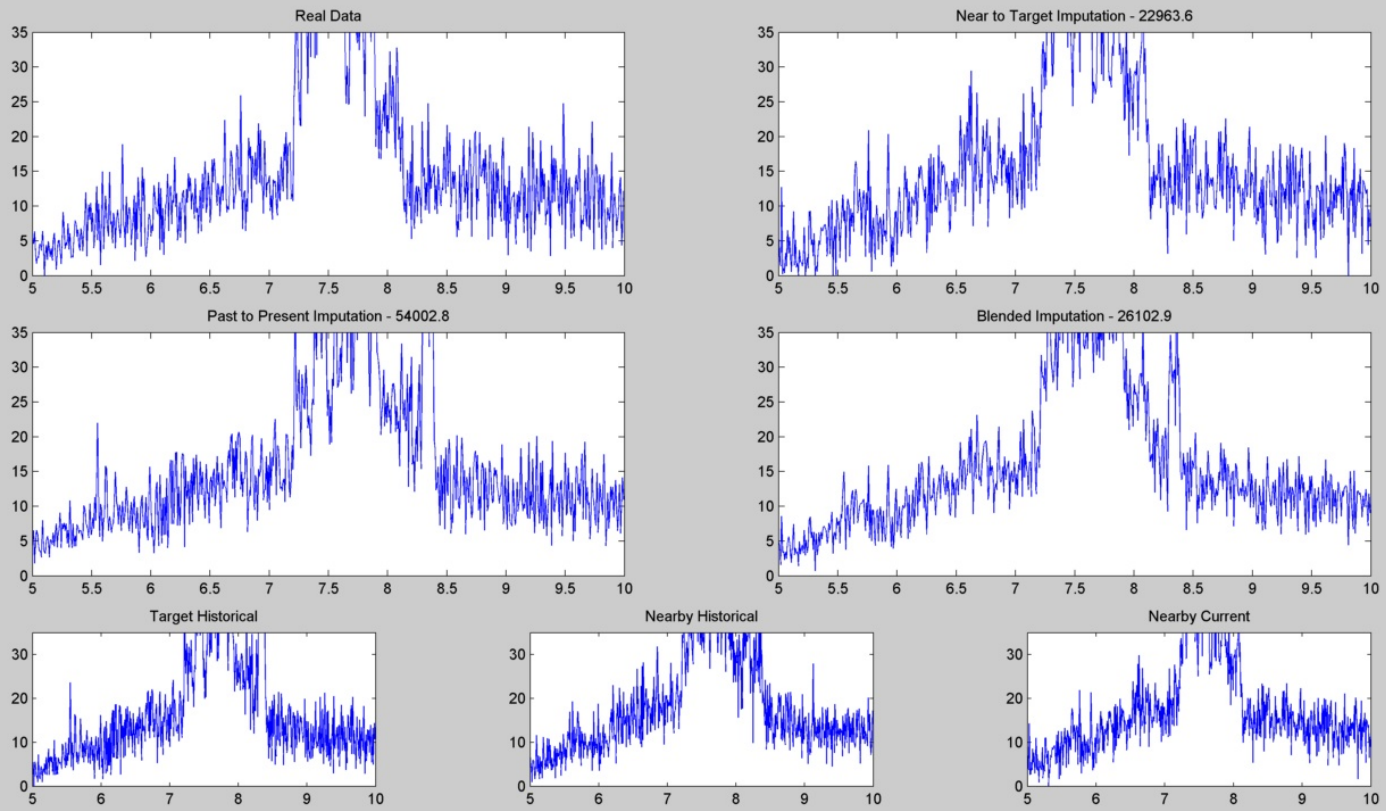


#	Target	Nearby	Current Date	Historical Date
3	512	513	10/13/2010	10/28/2010

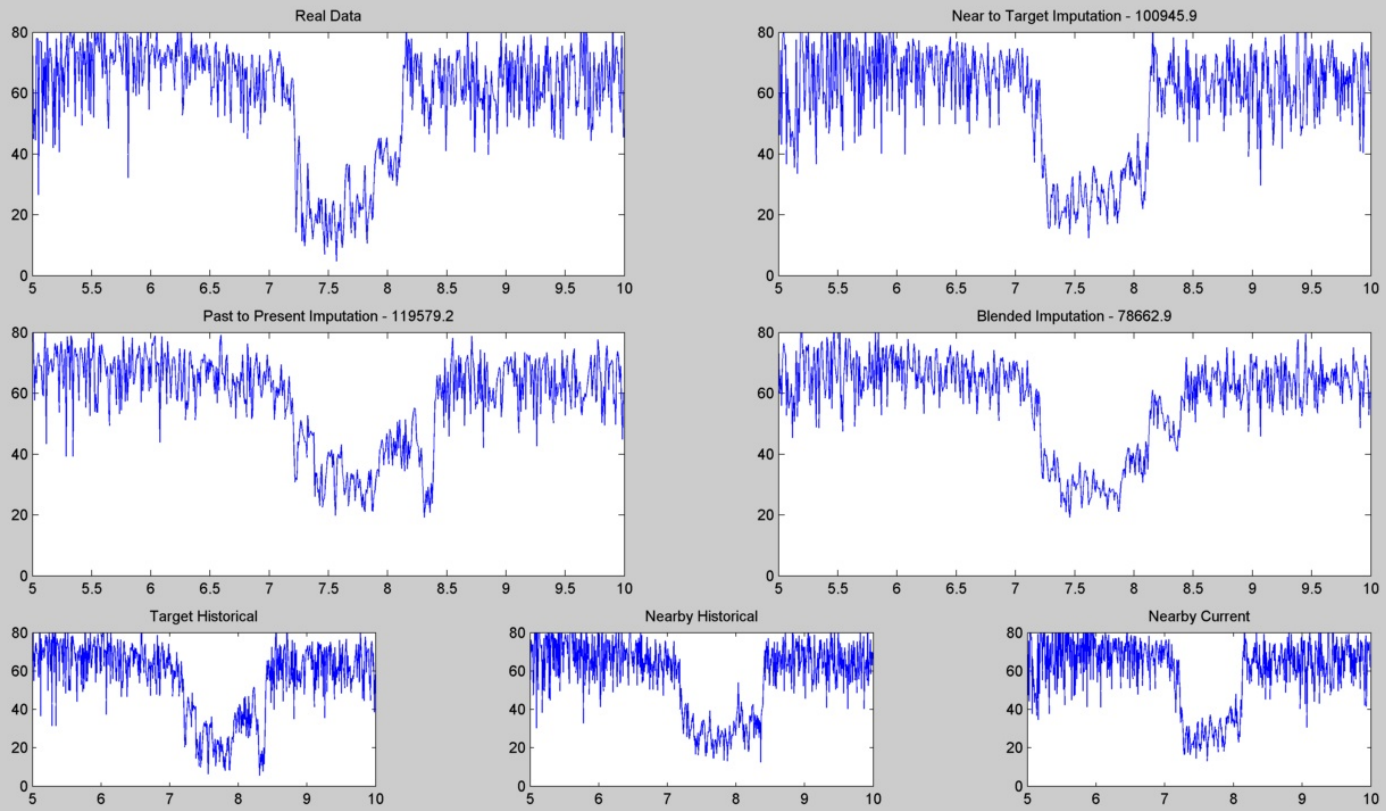
Volume - Correlation: 0.913



Occupancy - Correlation: 0.913

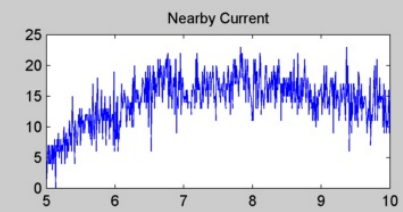
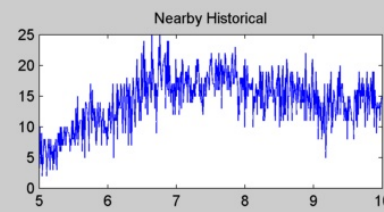
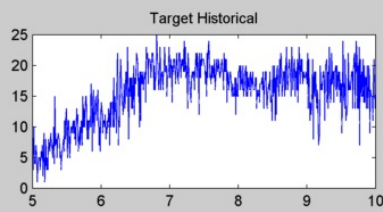
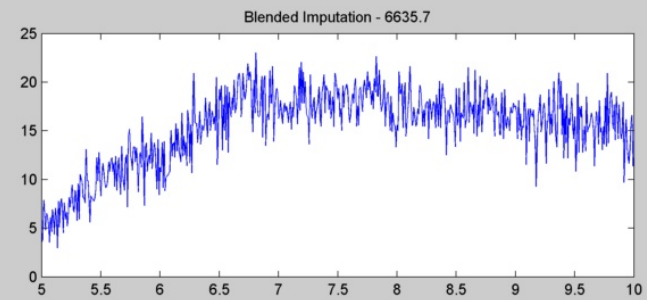
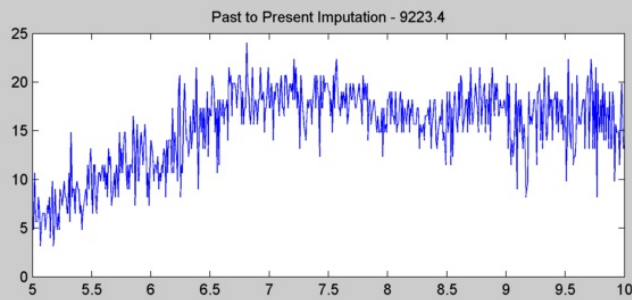
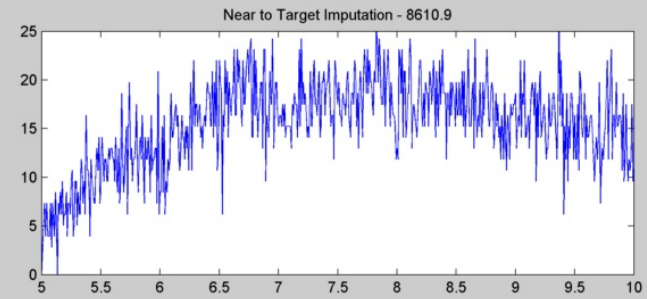
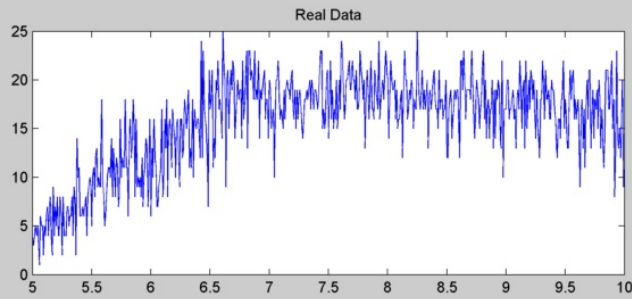


Speed - Correlation: 0.913

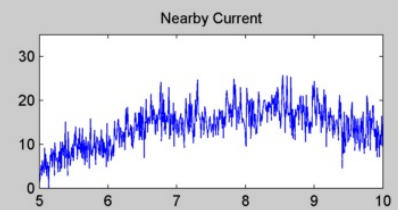
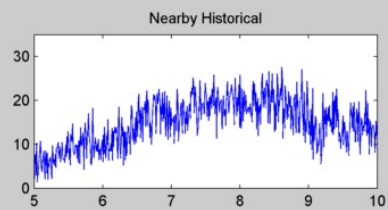
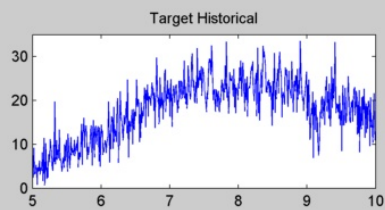
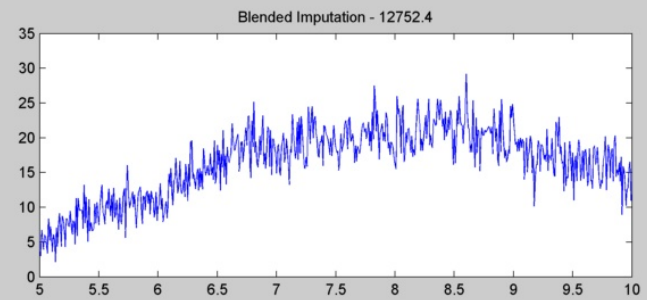
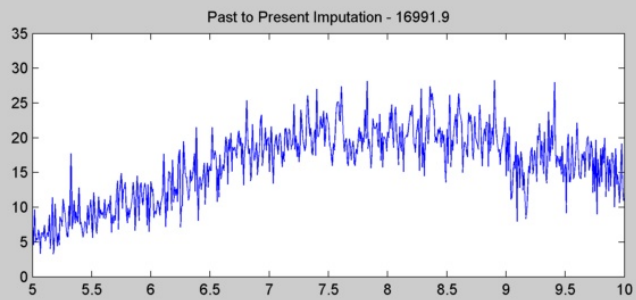
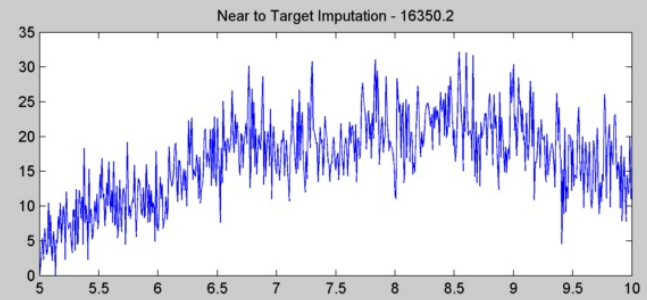
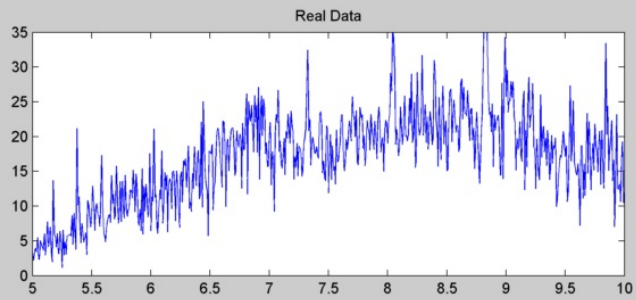


#	Target	Nearby	Current Date	Historical Date
4	843	844	10/12/2010	10/28/2010

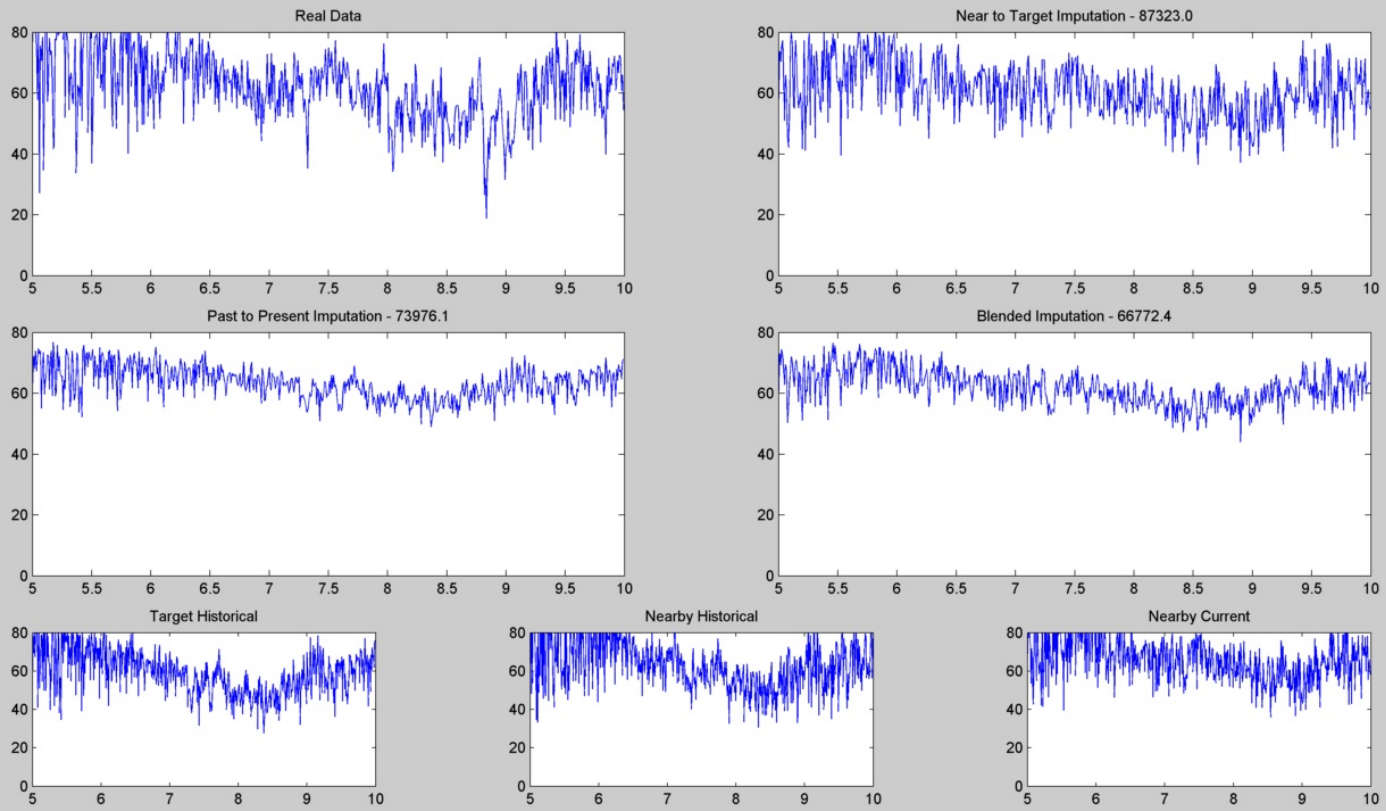
Volume - Correlation: 0.889



Occupancy - Correlation: 0.889

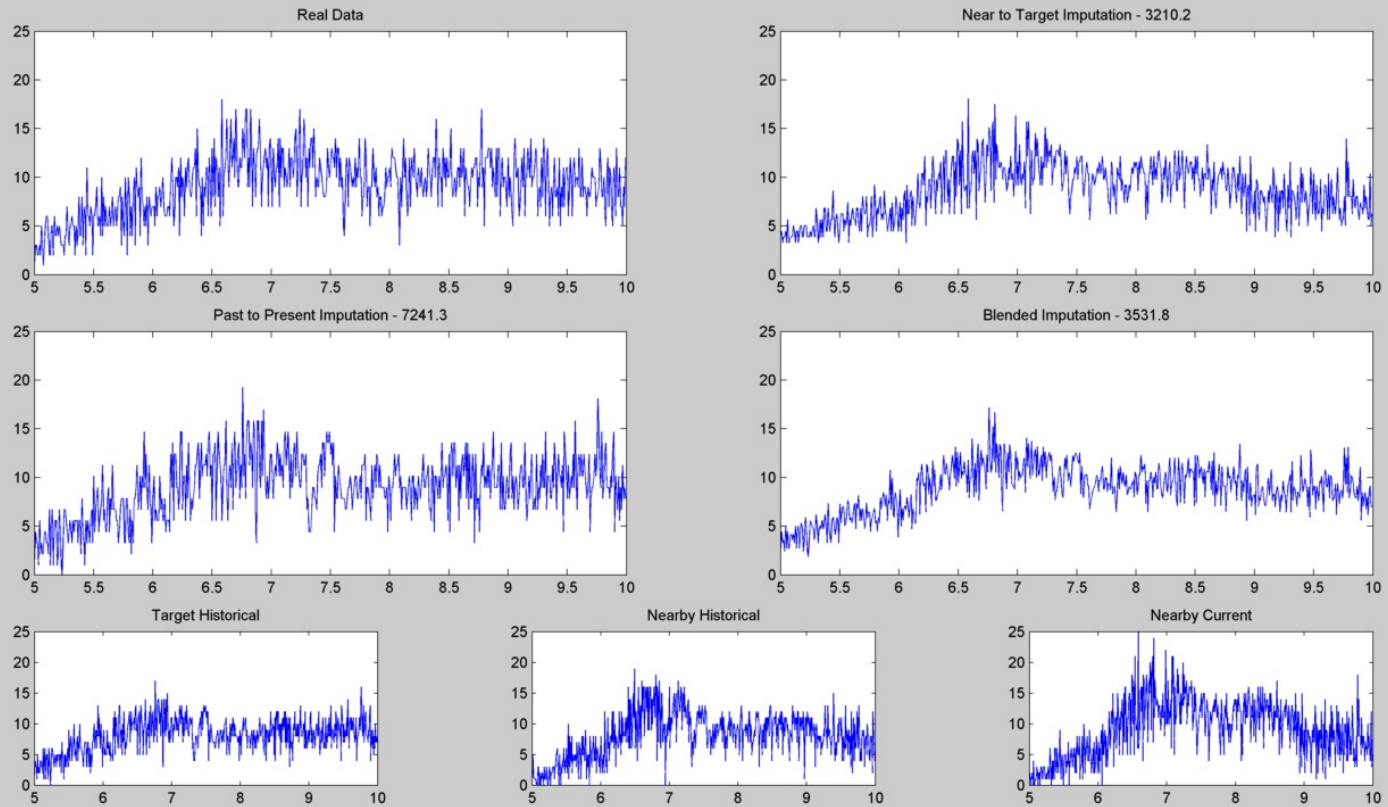


Speed - Correlation: 0.889

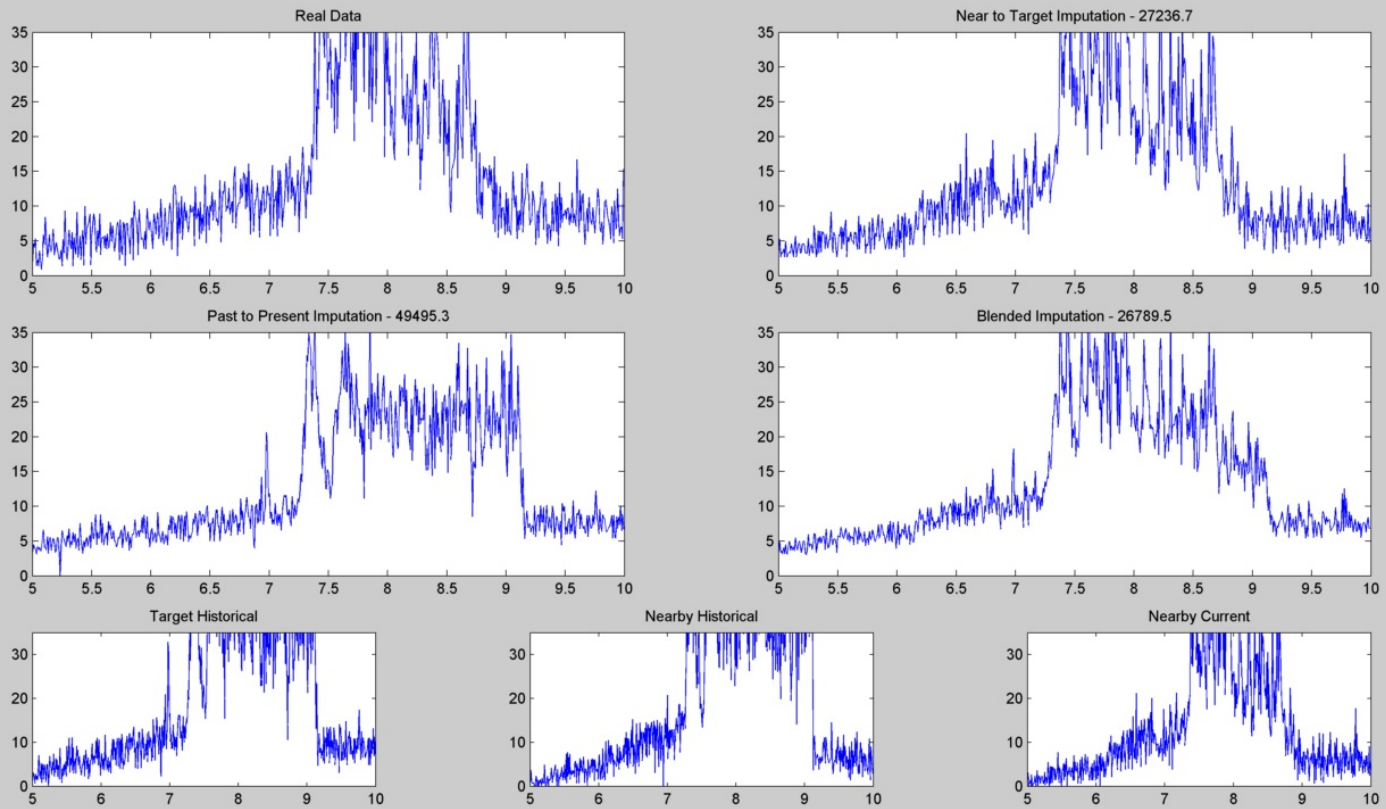


#	Target	Nearby	Current Date	Historical Date
5	1610	1611	10/12/2010	10/27/2010

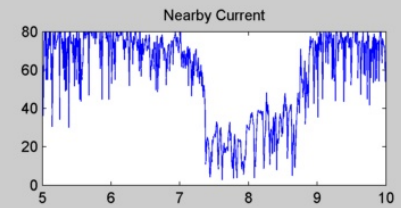
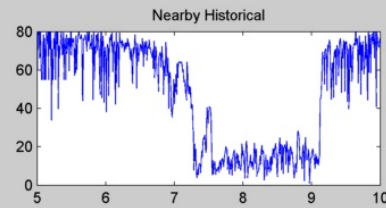
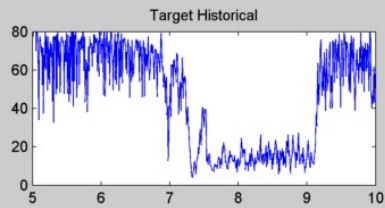
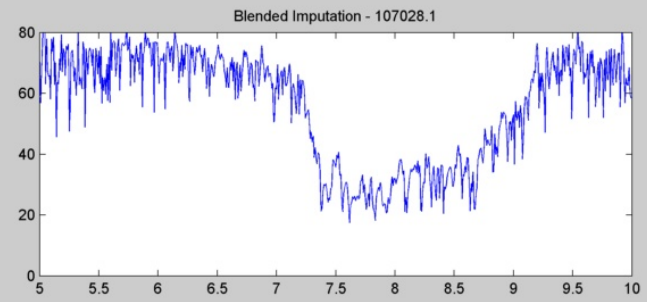
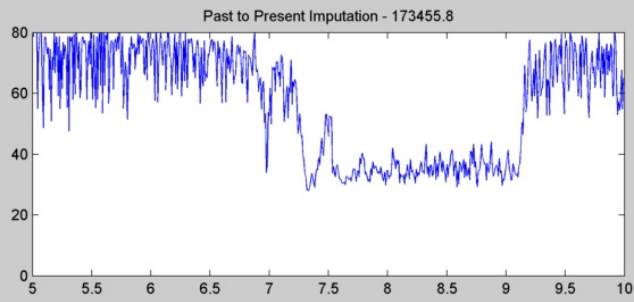
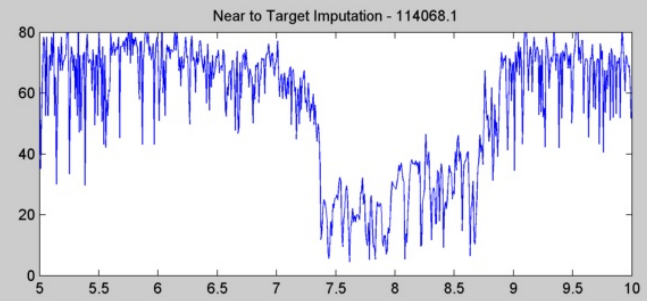
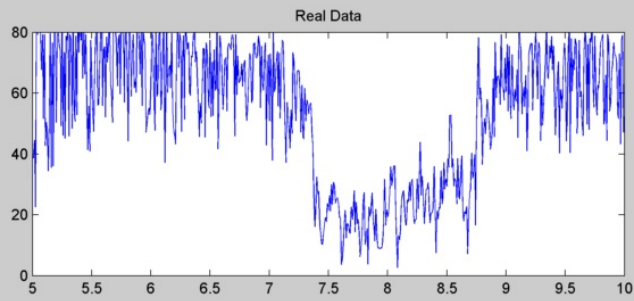
Volume - Correlation: 0.838



Occupancy - Correlation: 0.838

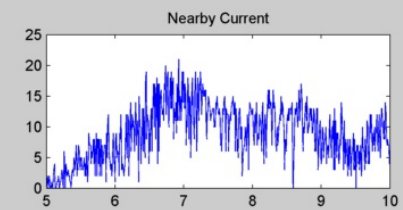
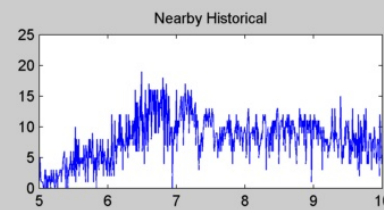
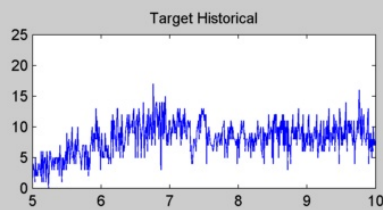
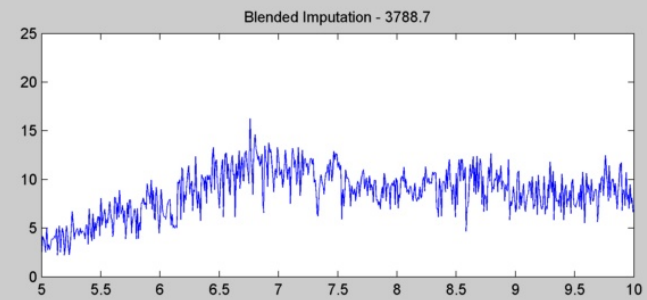
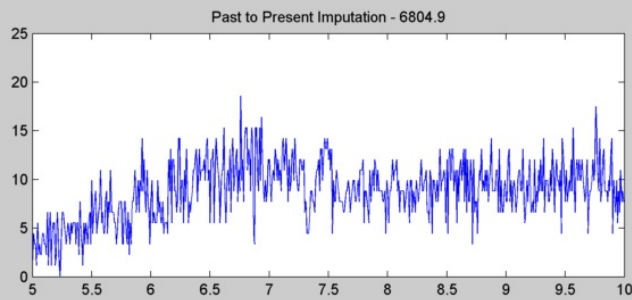
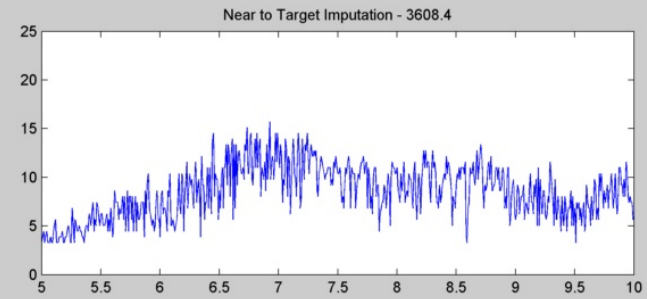
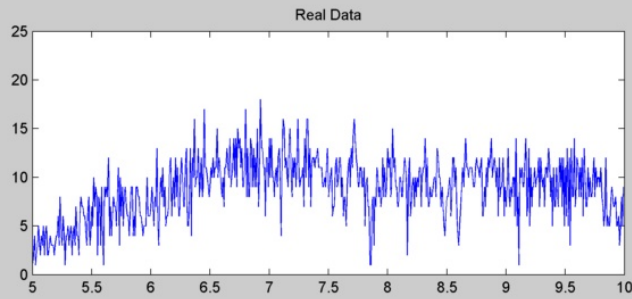


Speed - Correlation: 0.838

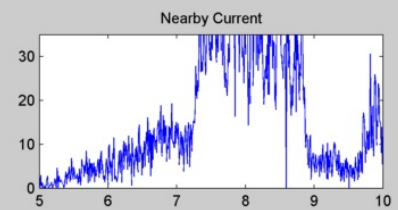
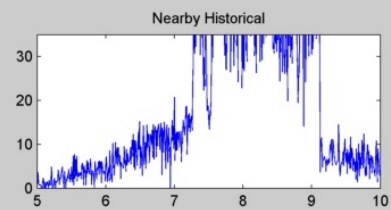
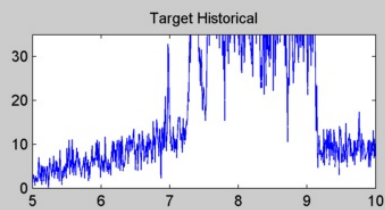
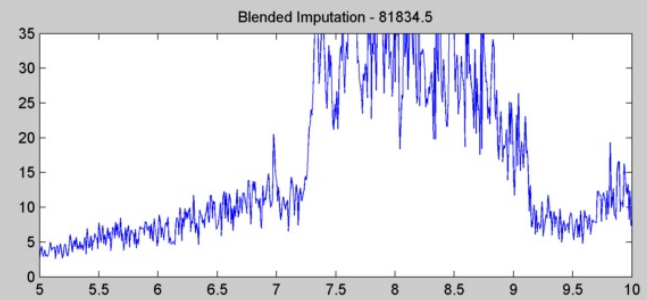
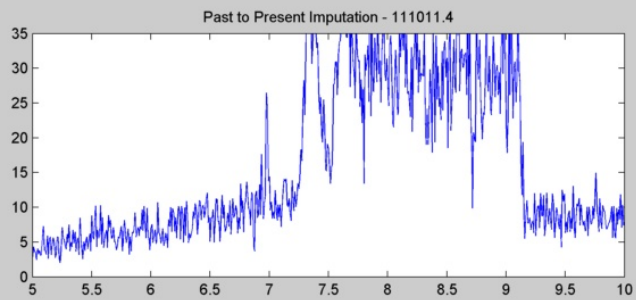
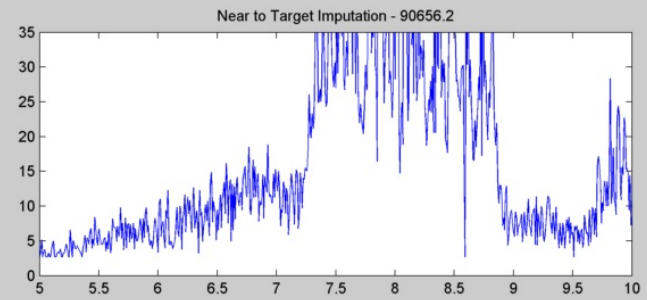
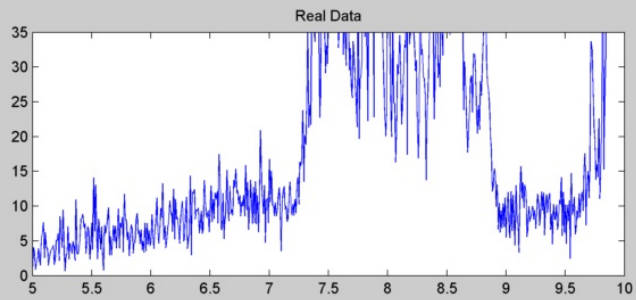


#	Target	Nearby	Current Date	Historical Date
6	1610	1611	10/14/2010	10/27/2010

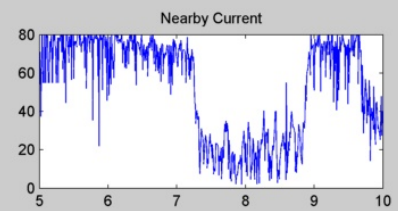
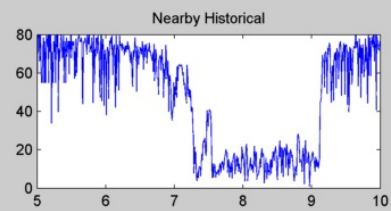
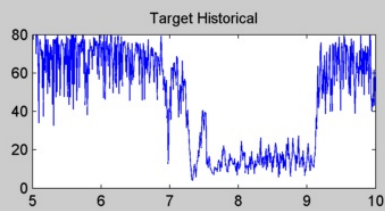
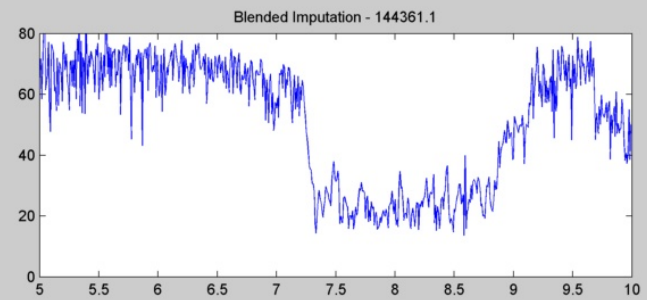
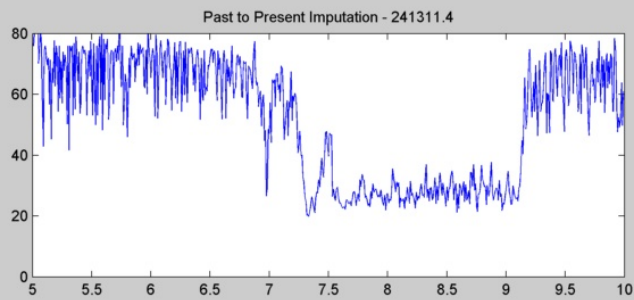
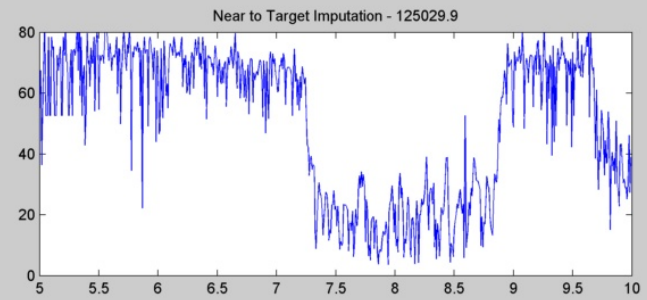
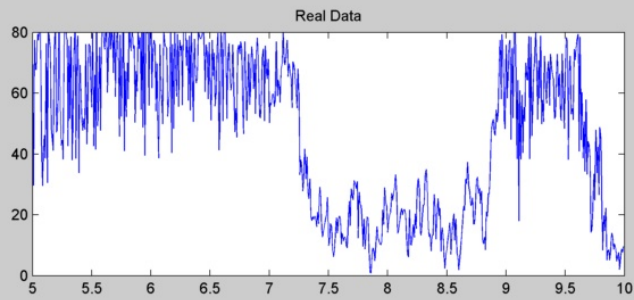
Volume - Correlation: 0.883



Occupancy - Correlation: 0.883

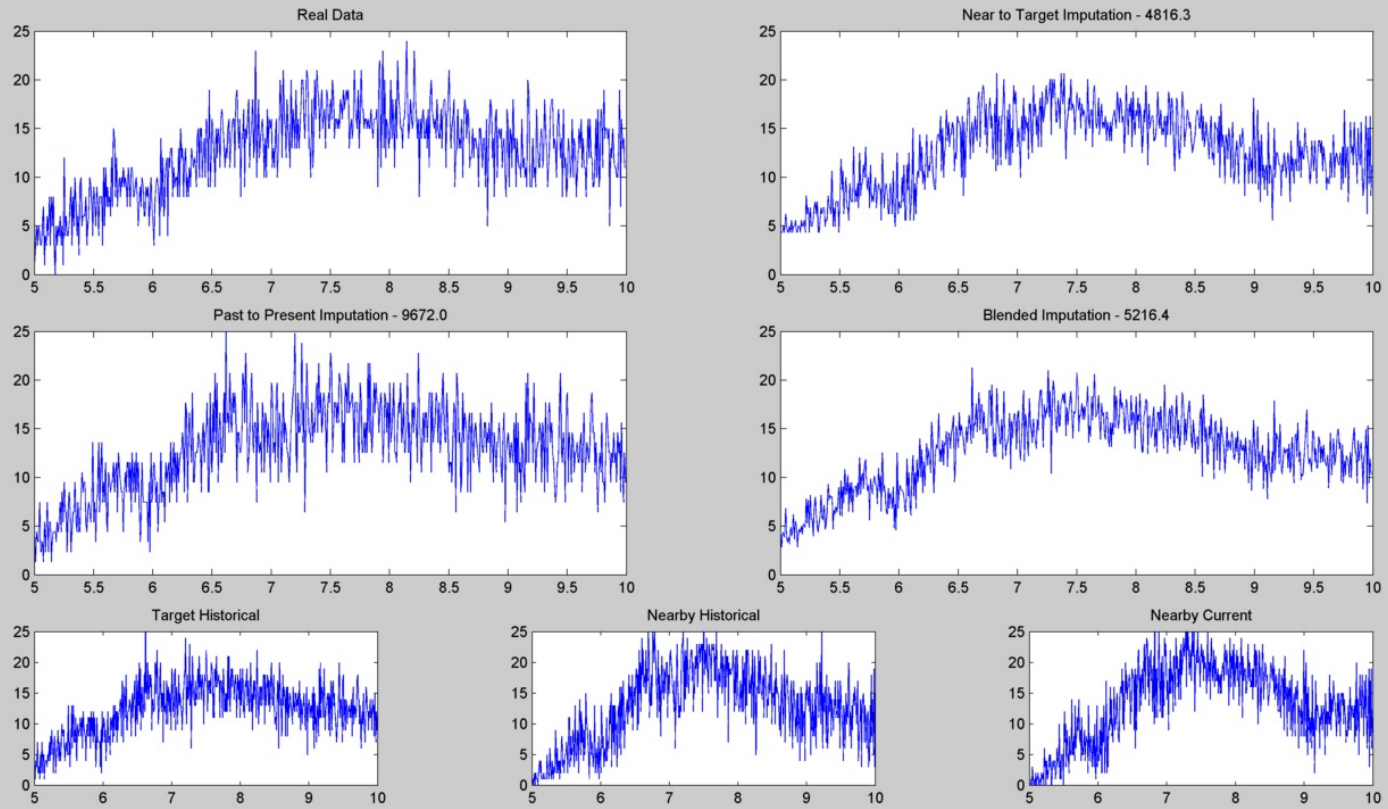


Speed - Correlation: 0.883

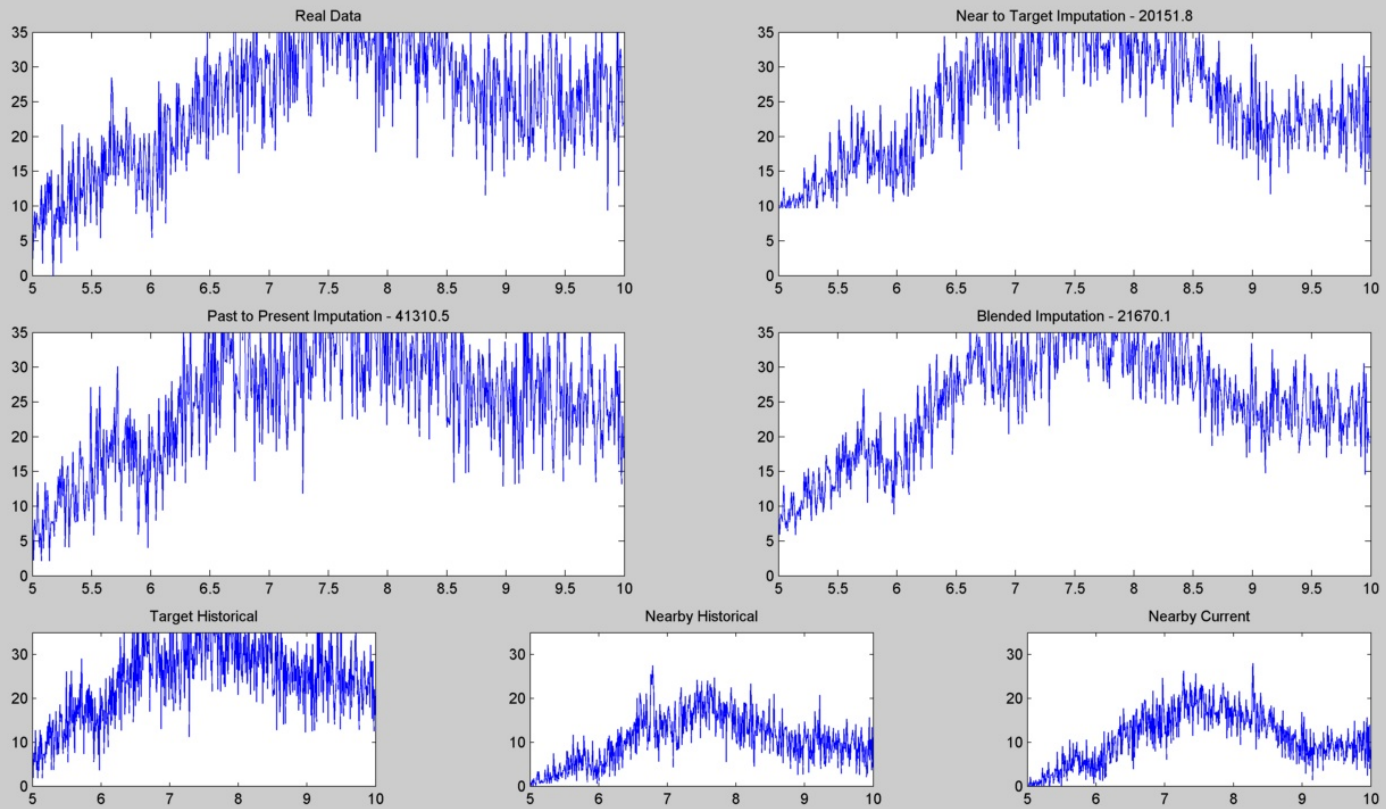


#	Target	Nearby	Current Date	Historical Date
7	2669	2670	10/14/2010	10/21/2010

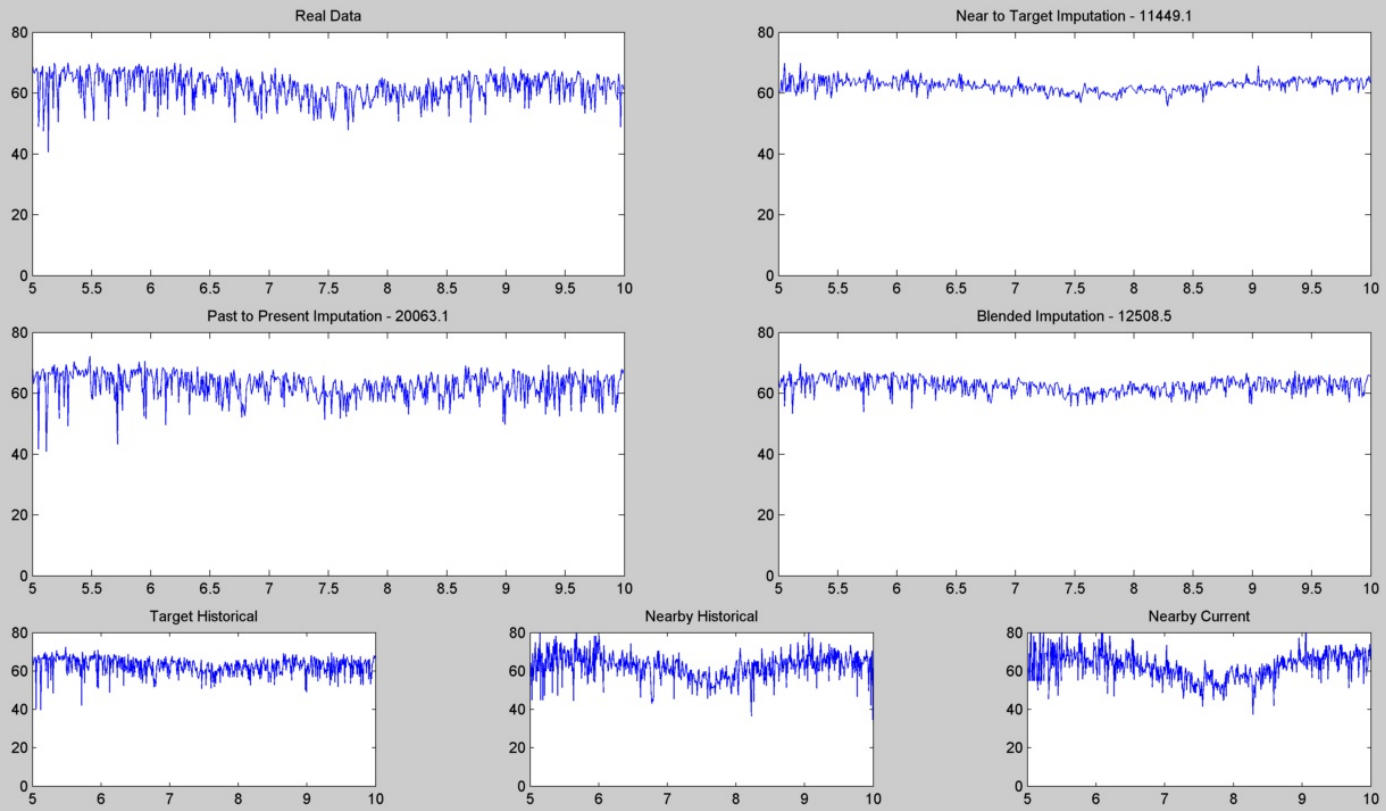
Volume - Correlation: 0.936



Occupancy - Correlation: 0.936

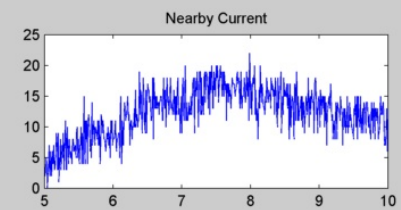
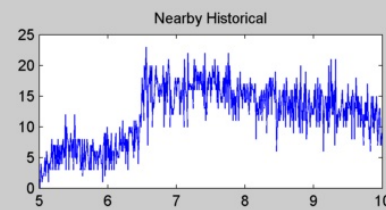
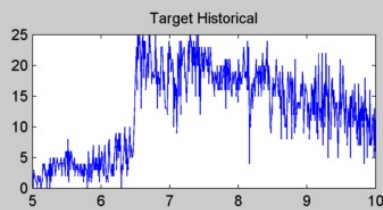
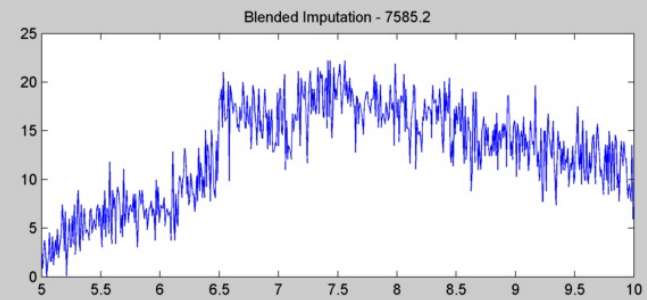
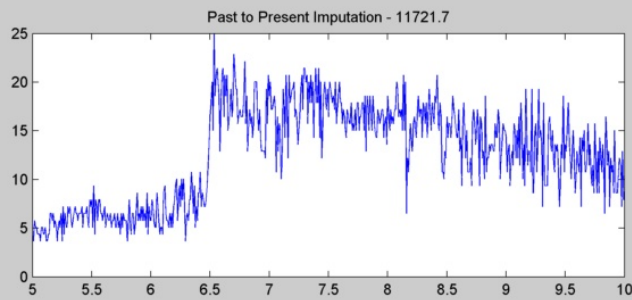
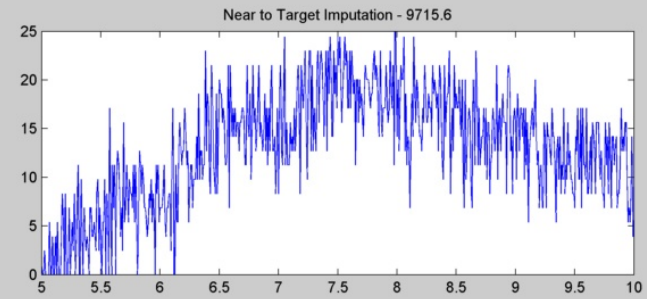
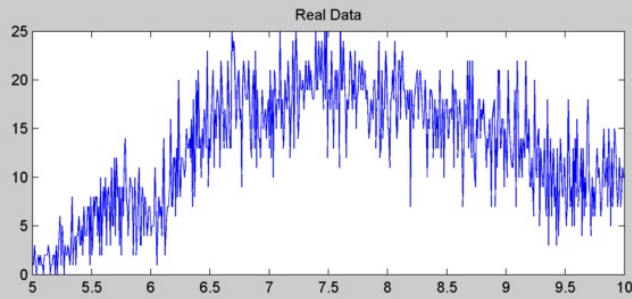


Speed - Correlation: 0.936

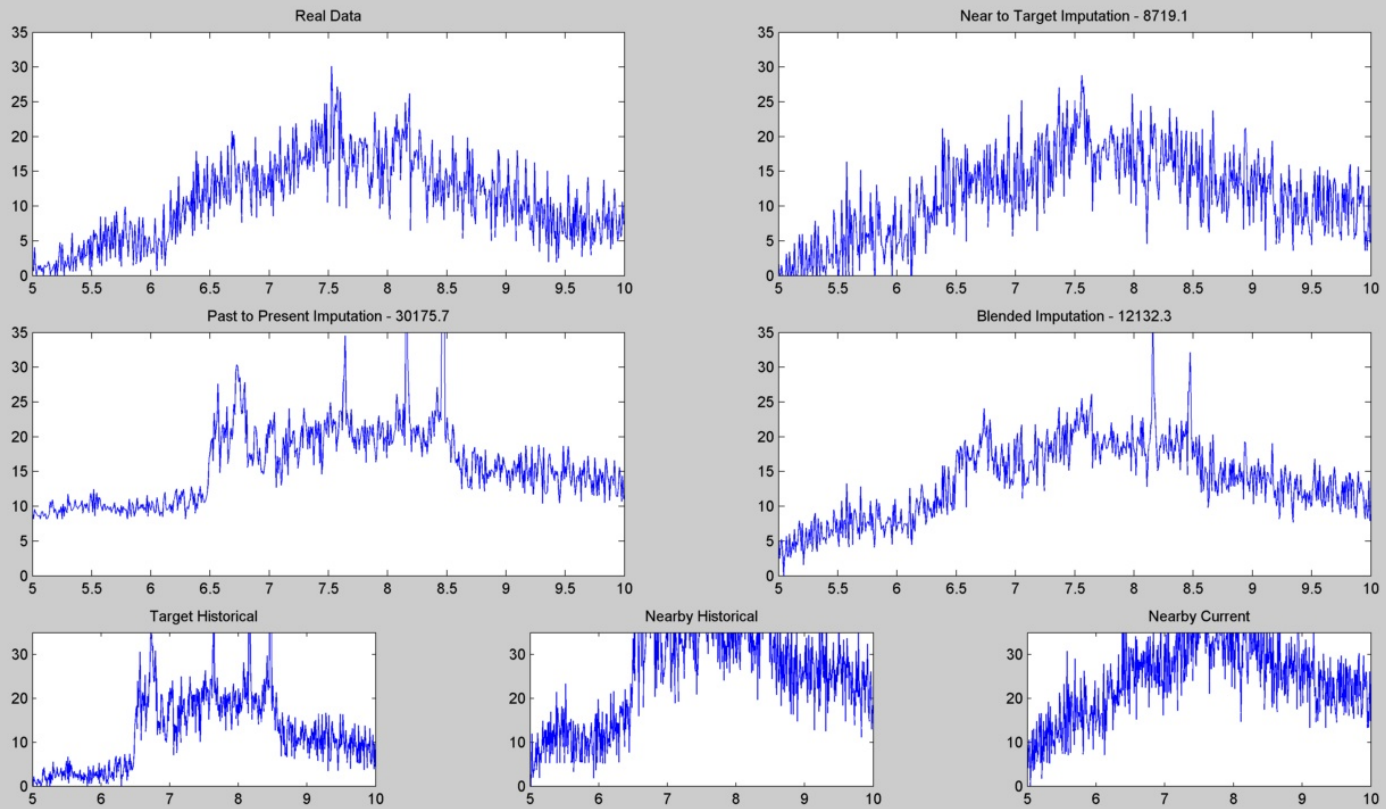


#	Target	Nearby	Current Date	Historical Date
8	2670	2669	10/19/2010	10/28/2010

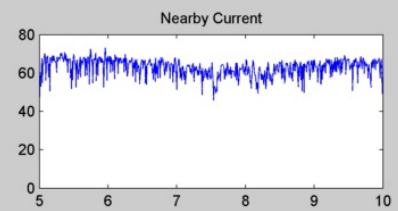
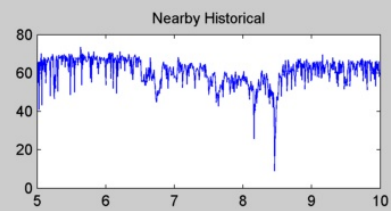
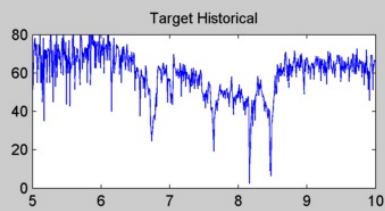
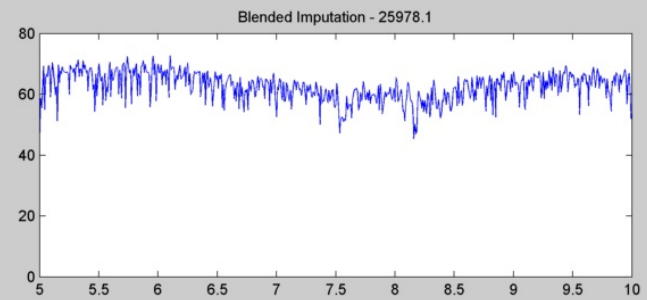
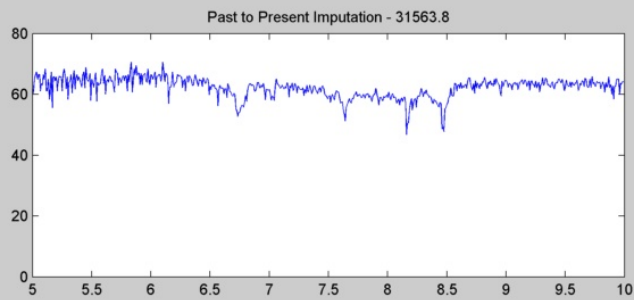
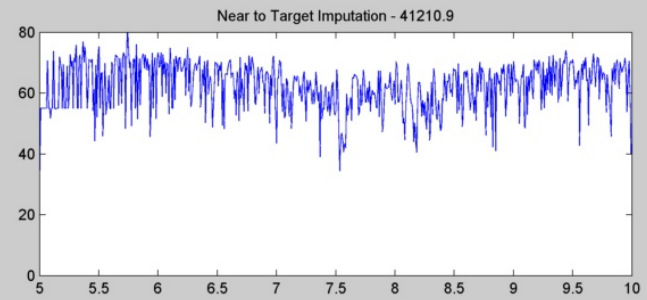
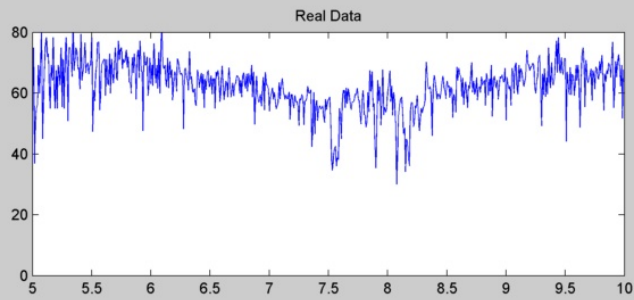
Volume - Correlation: 0.890



Occupancy - Correlation: 0.890

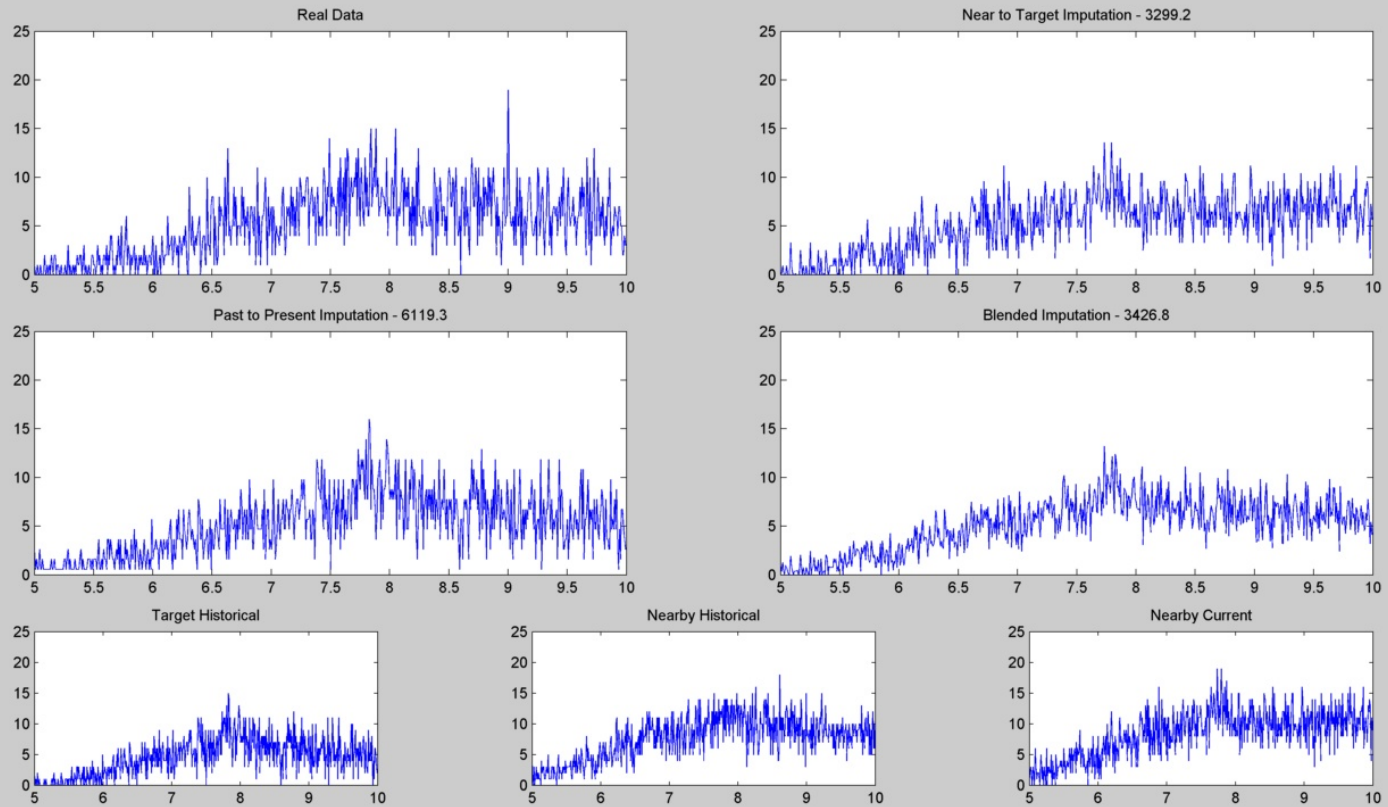


Speed - Correlation: 0.890

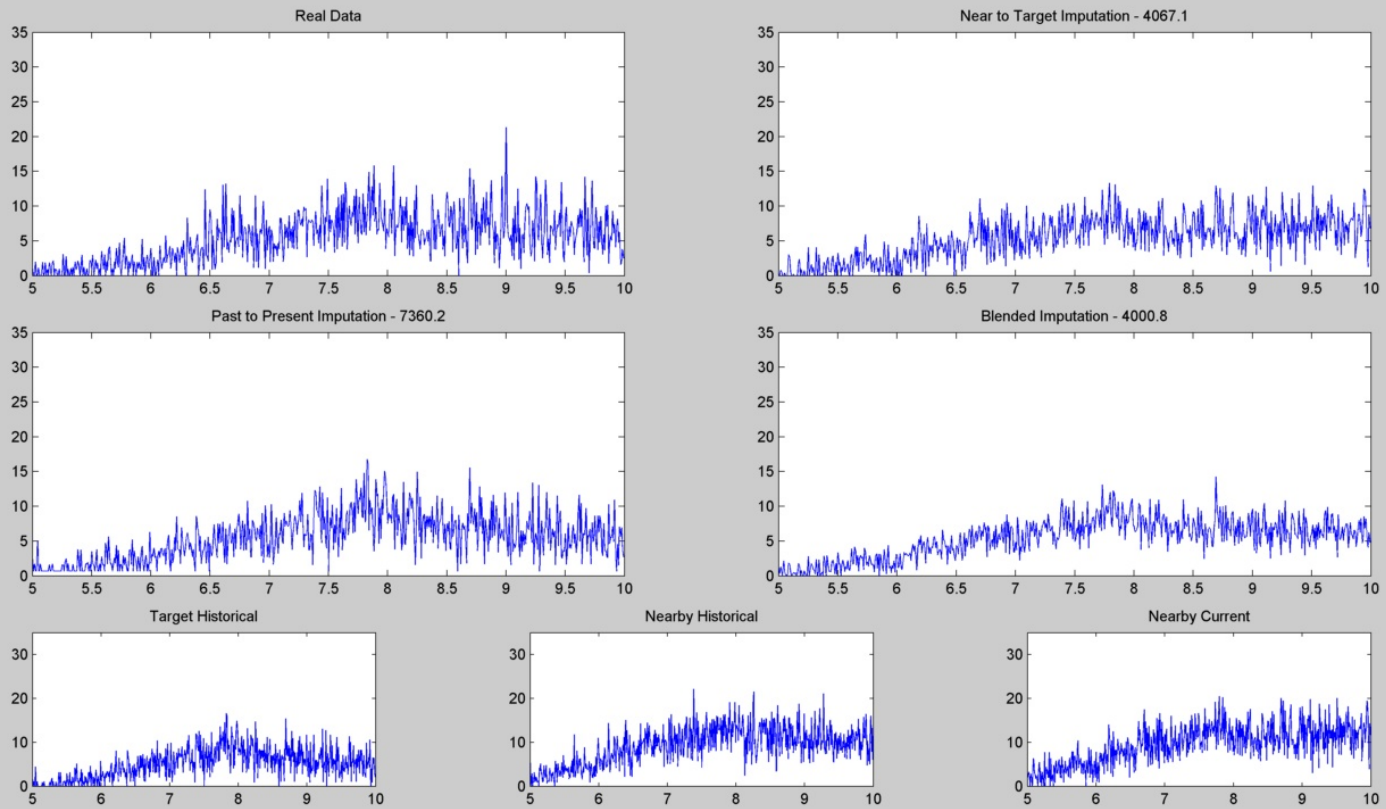


#	Target	Nearby	Current Date	Historical Date
9	3190	3189	10/21/2010	10/27/2010

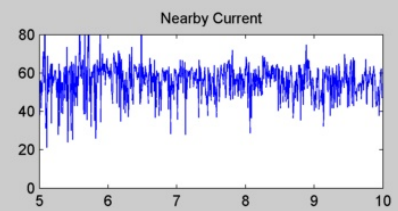
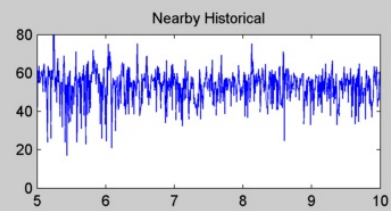
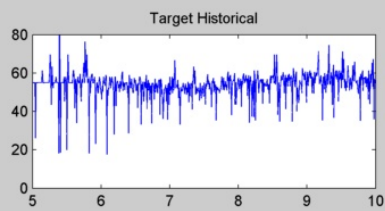
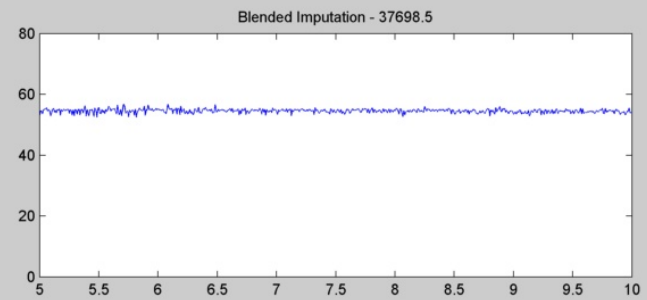
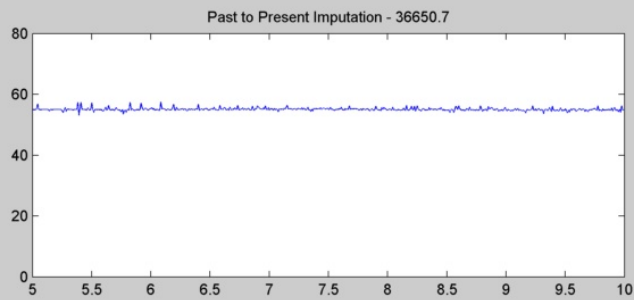
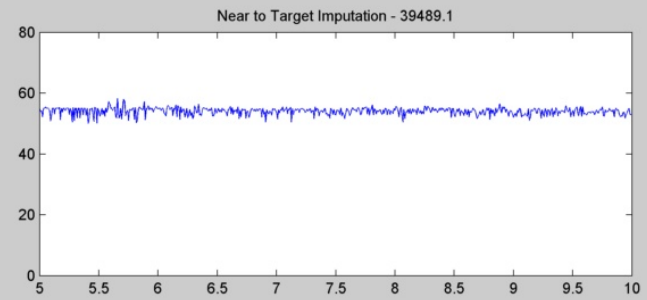
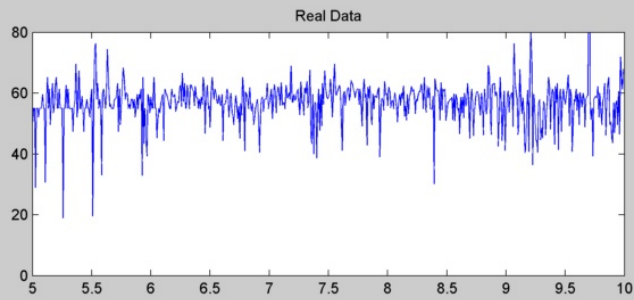
Volume - Correlation: 0.922



Occupancy - Correlation: 0.922

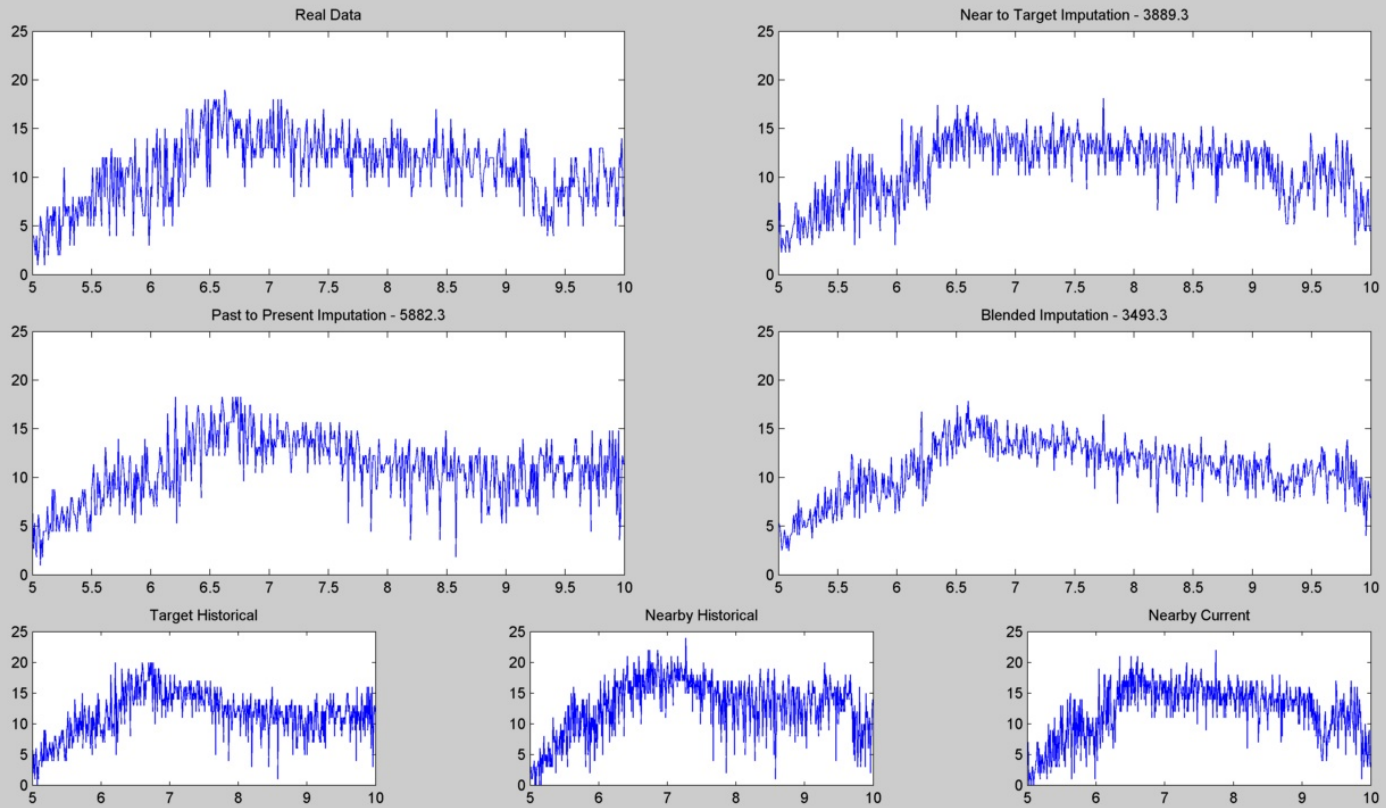


Speed - Correlation: 0.922

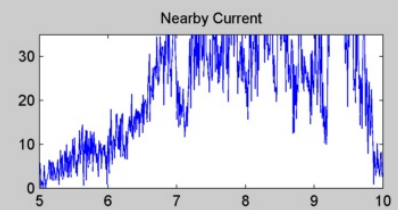
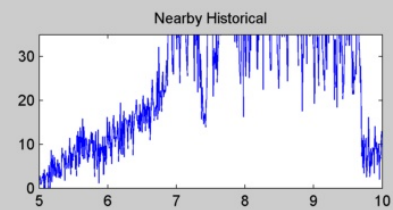
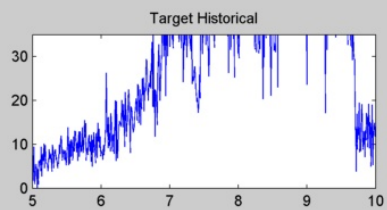
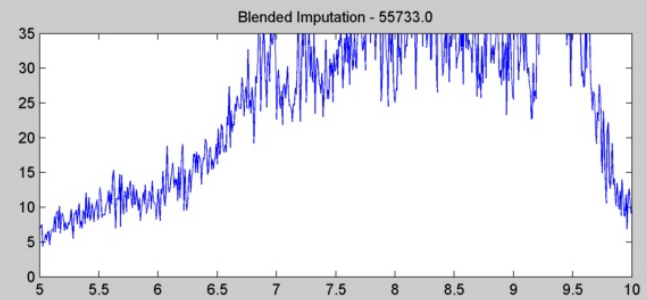
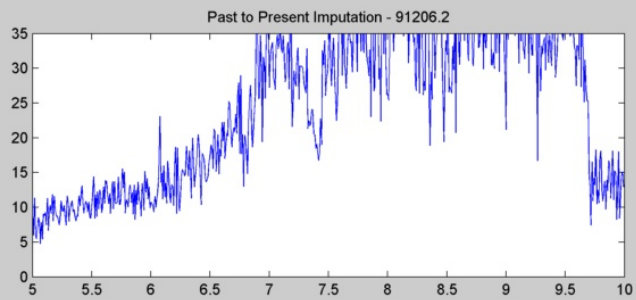
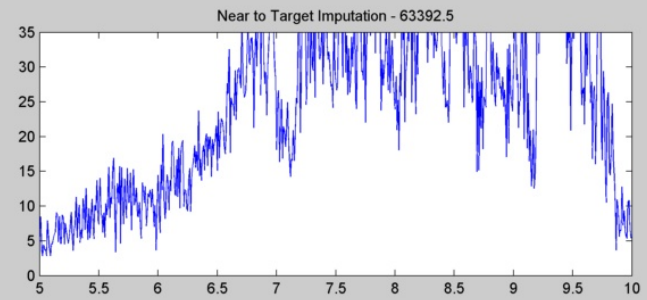
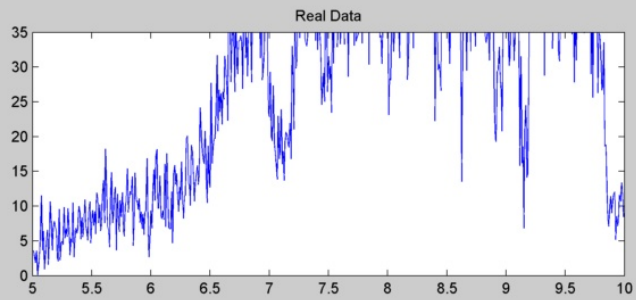


#	Target	Nearby	Current Date	Historical Date
10	3235	3236	10/26/2010	9/30/2010

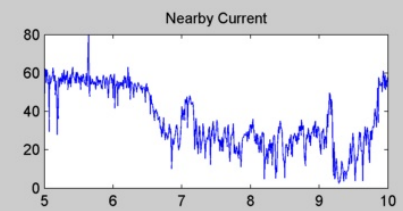
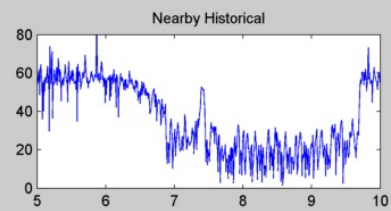
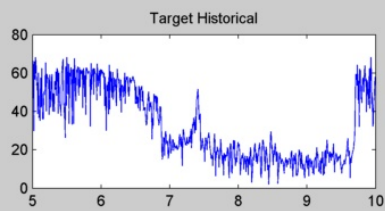
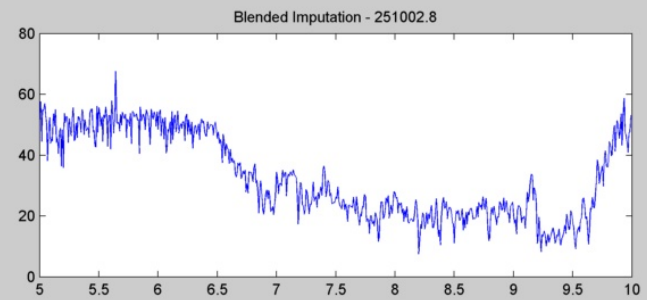
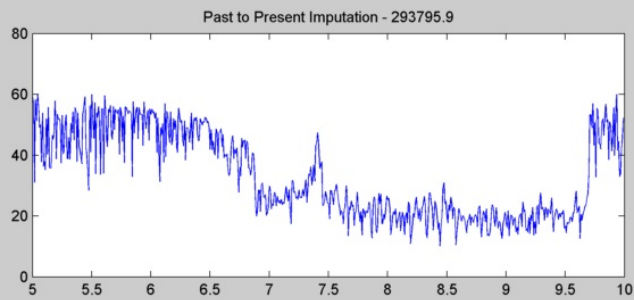
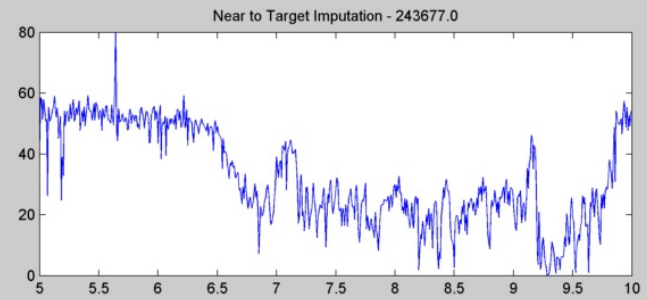
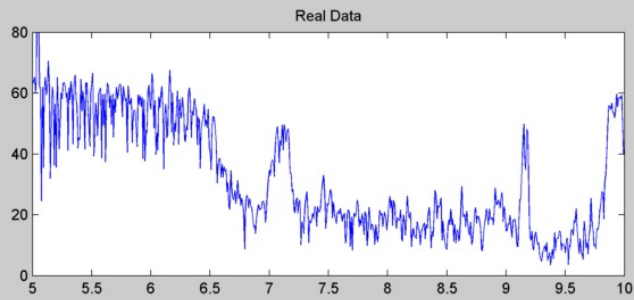
Volume - Correlation: 0.819



Occupancy - Correlation: 0.819

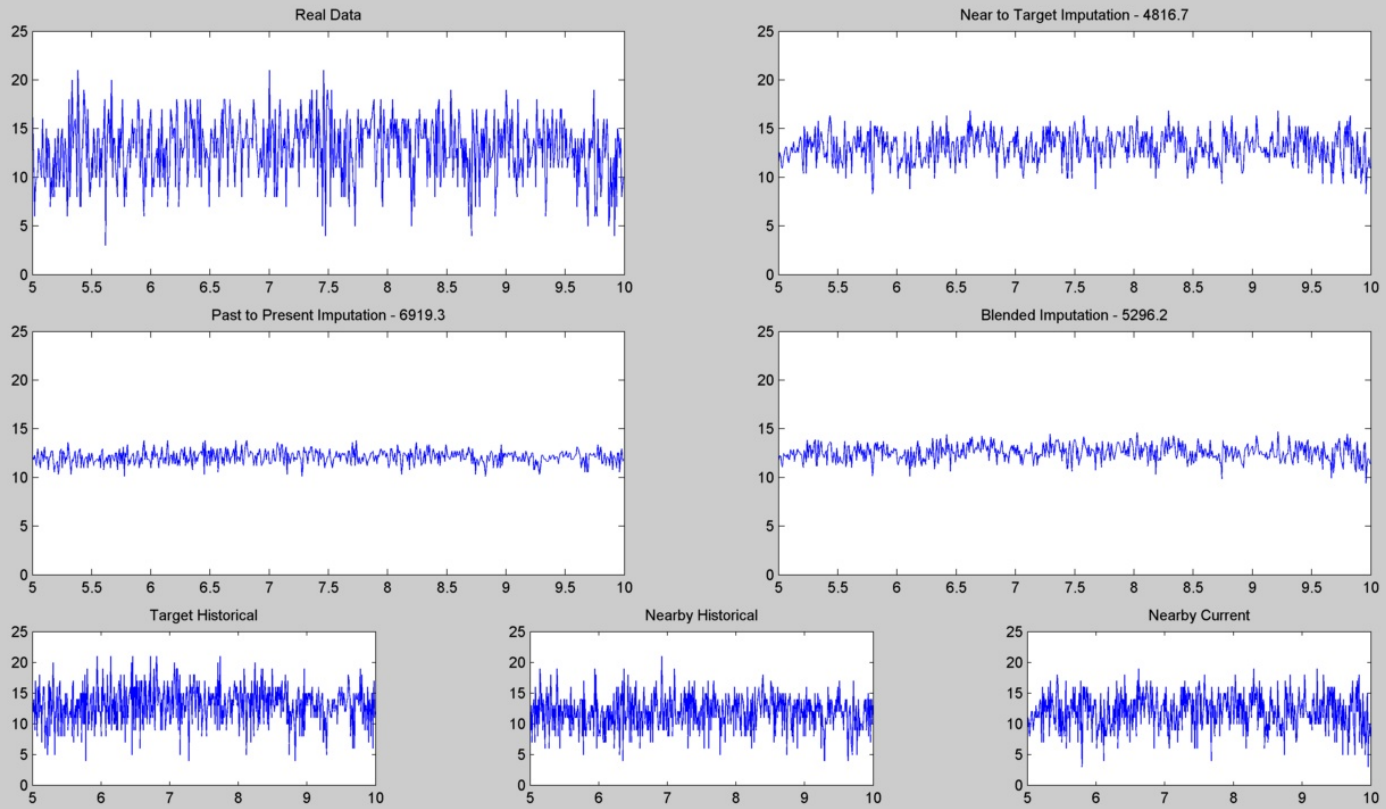


Speed - Correlation: 0.819

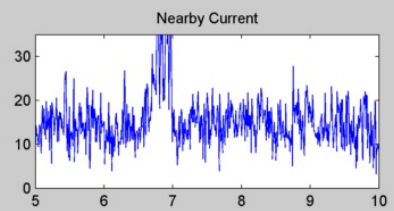
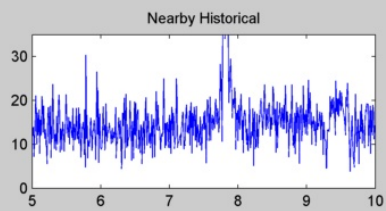
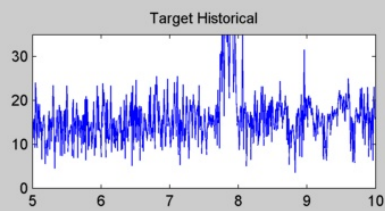
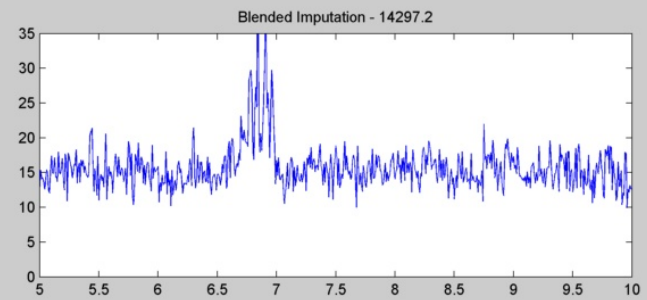
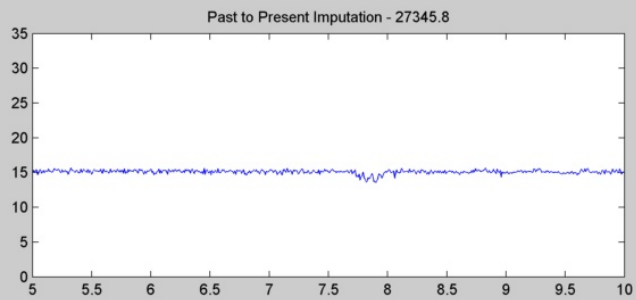
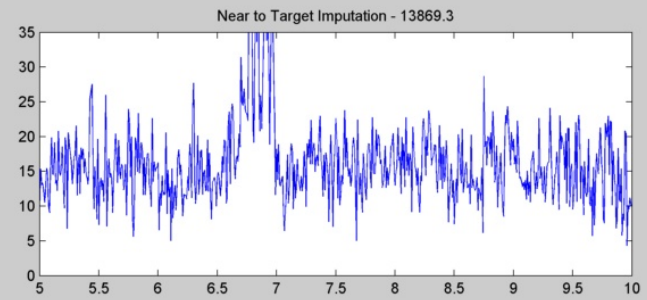
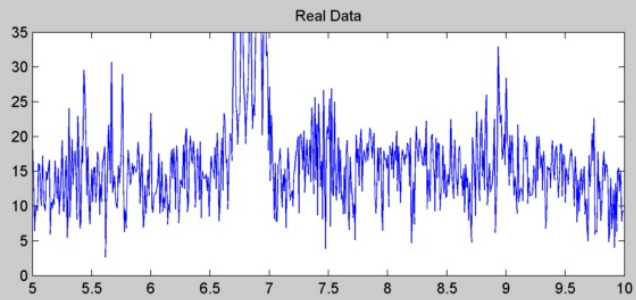


#	Target	Nearby	Current Date	Historical Date
11	340	339	10/20/2010	10/21/2010

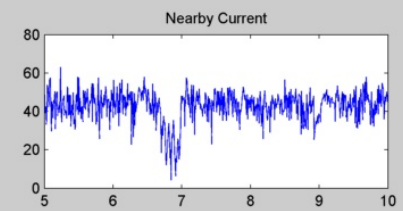
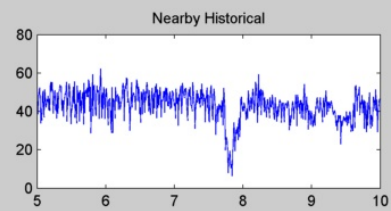
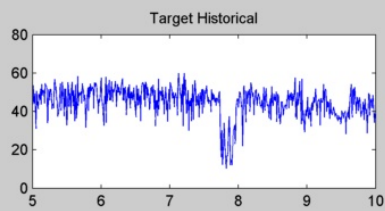
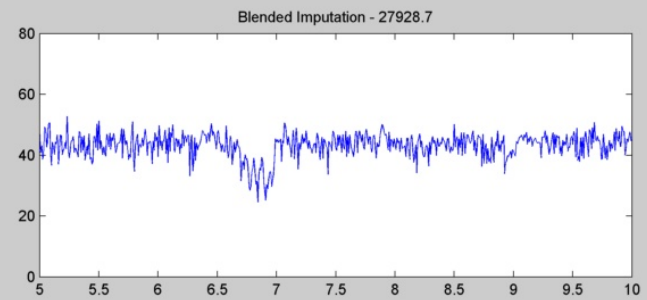
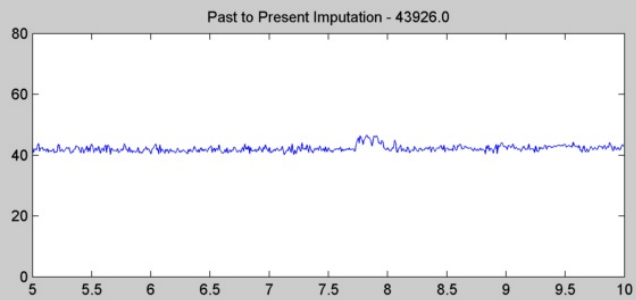
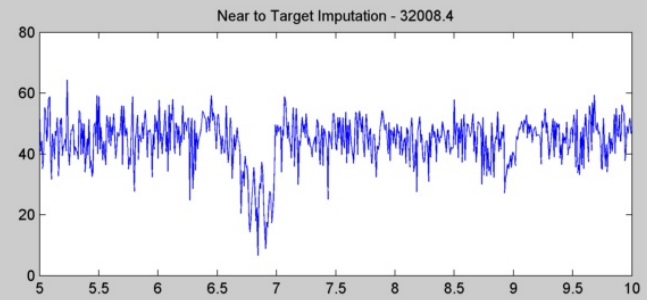
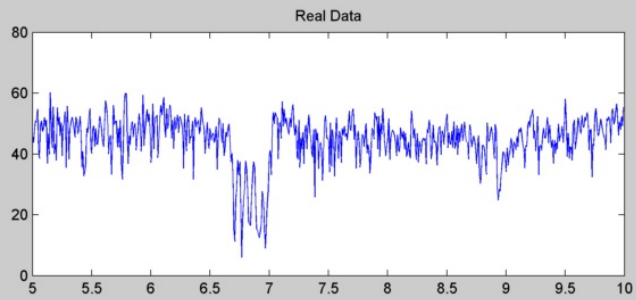
Volume - Correlation: 0.758



Occupancy - Correlation: 0.758

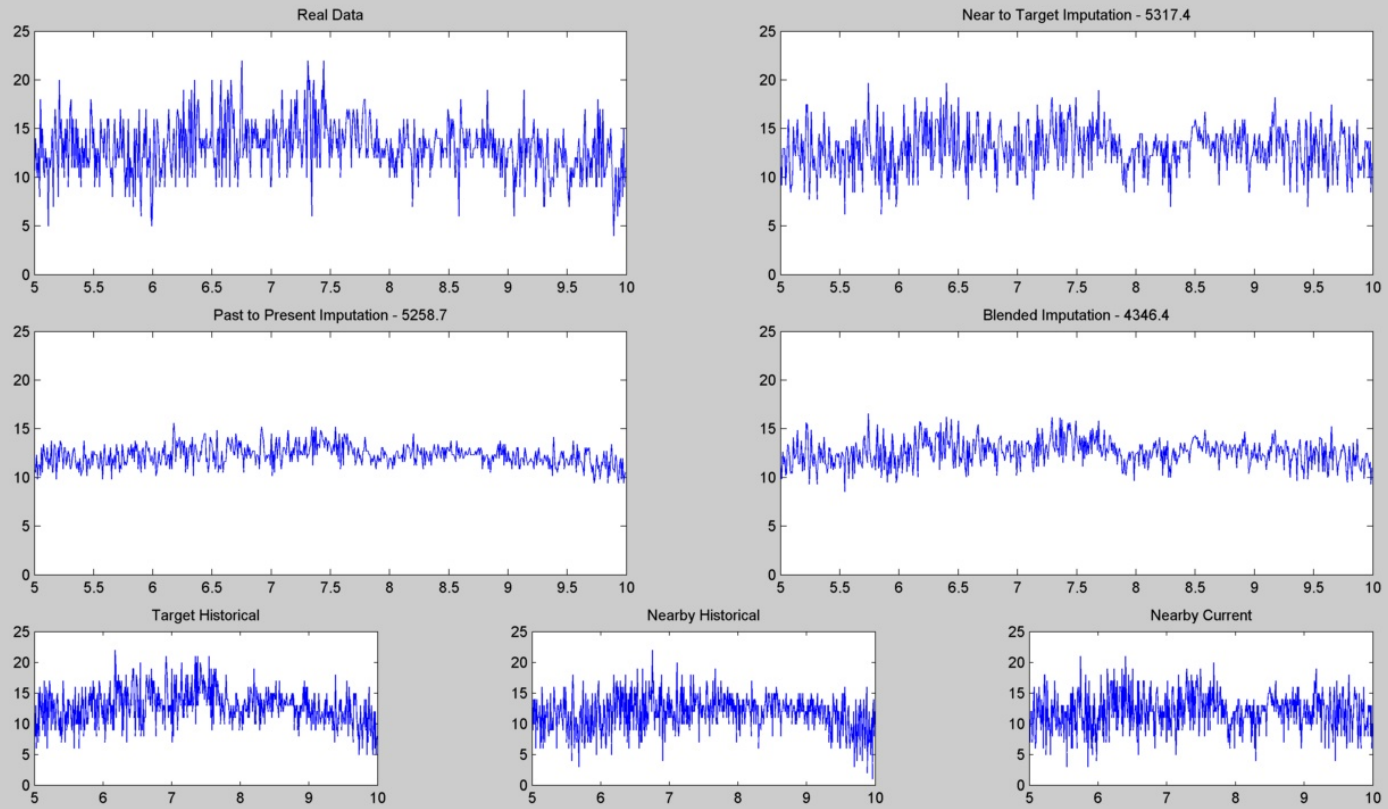


Speed - Correlation: 0.758

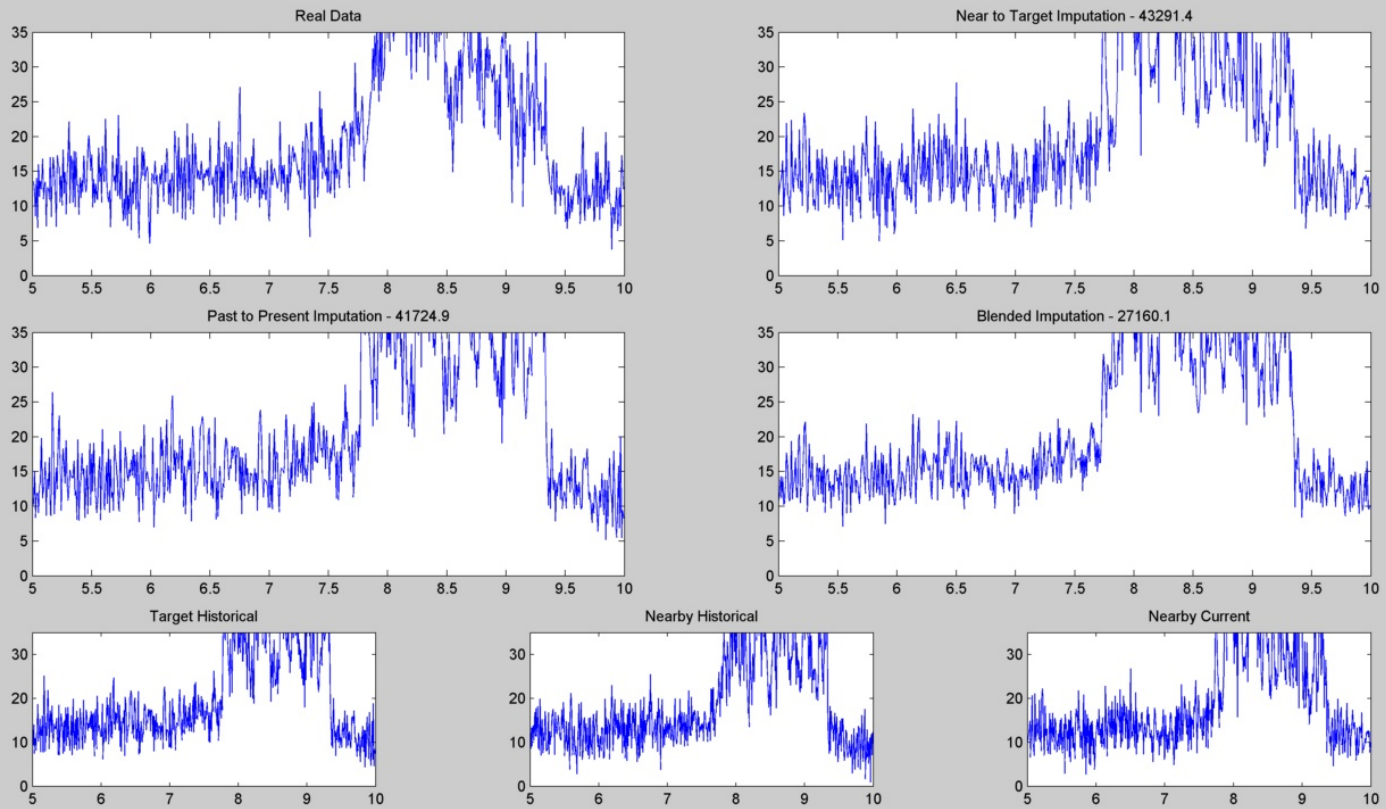


#	Target	Nearby	Current Date	Historical Date
12	513	512	10/21/2010	10/14/2010

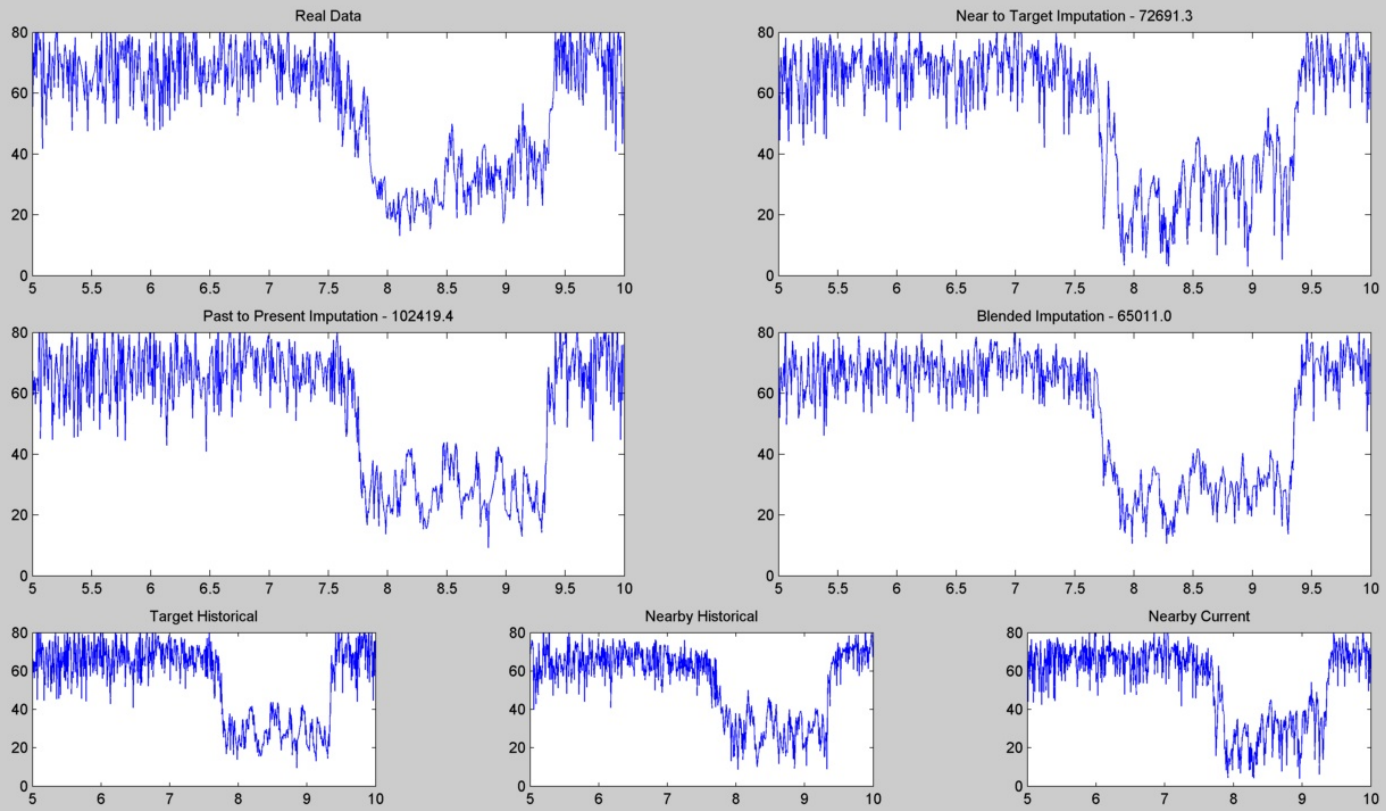
Volume - Correlation: 0.887



Occupancy - Correlation: 0.887

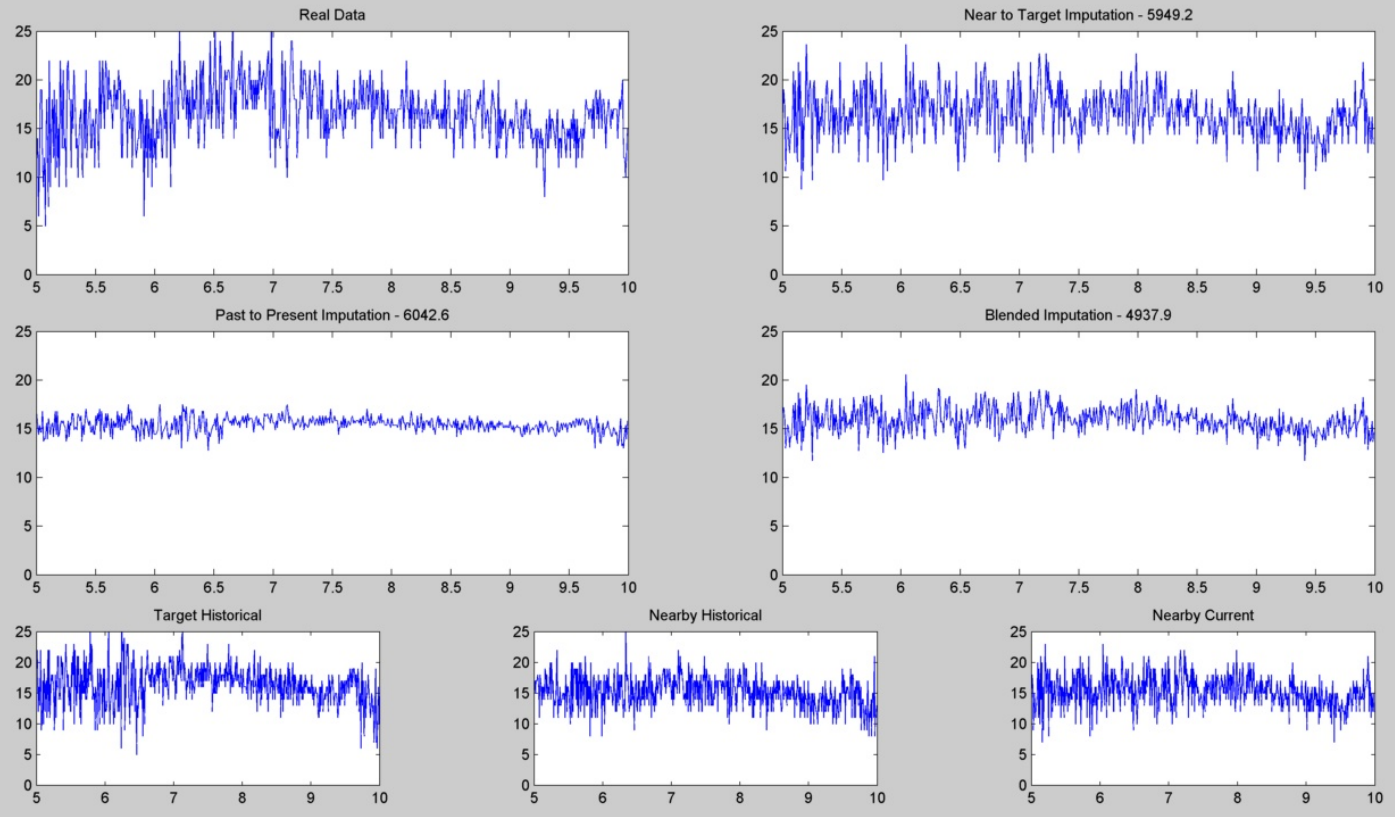


Speed - Correlation: 0.887

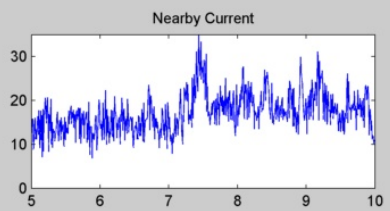
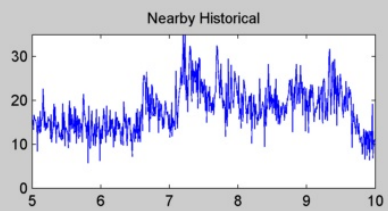
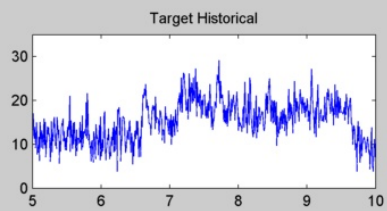
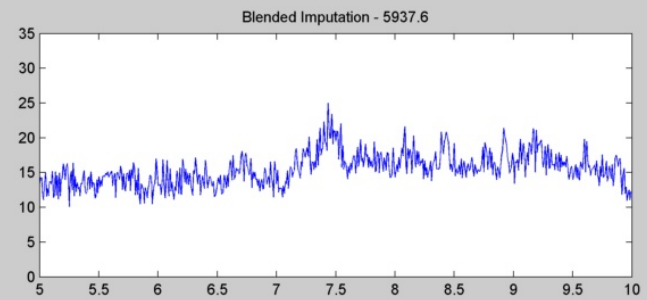
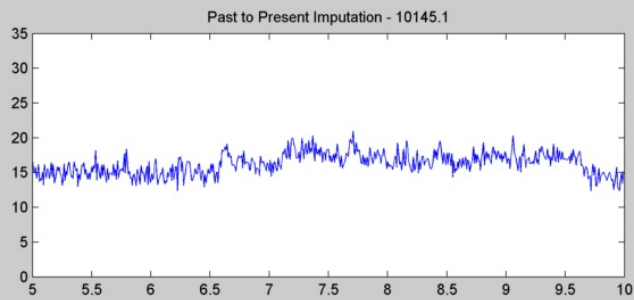
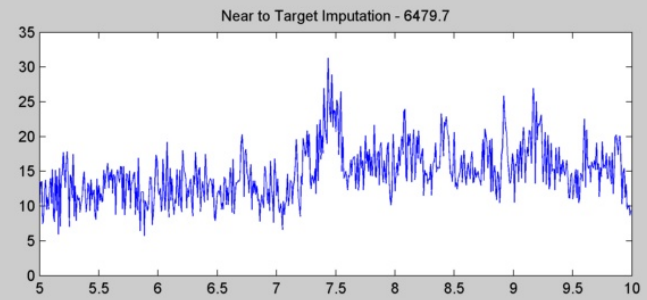
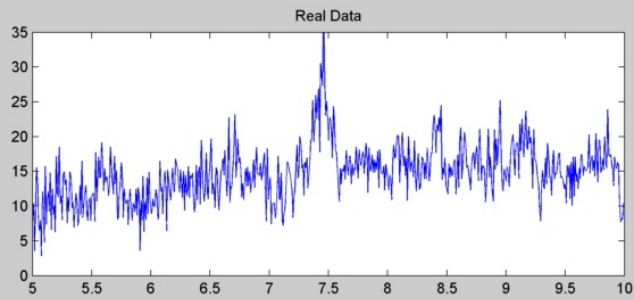


#	Target	Nearby	Current Date	Historical Date
13	845	844	10/5/2010	10/12/2010

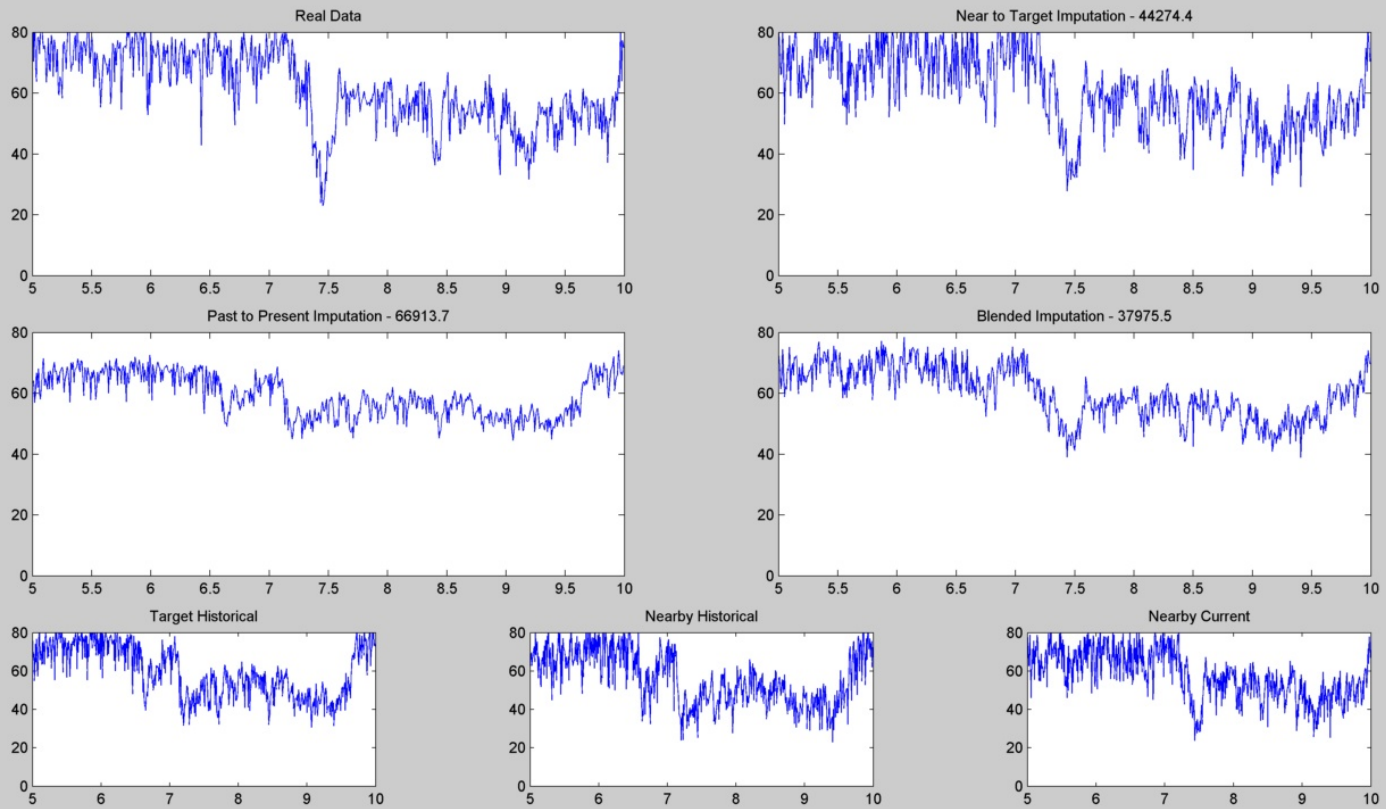
Volume - Correlation: 0.612



Occupancy - Correlation: 0.612

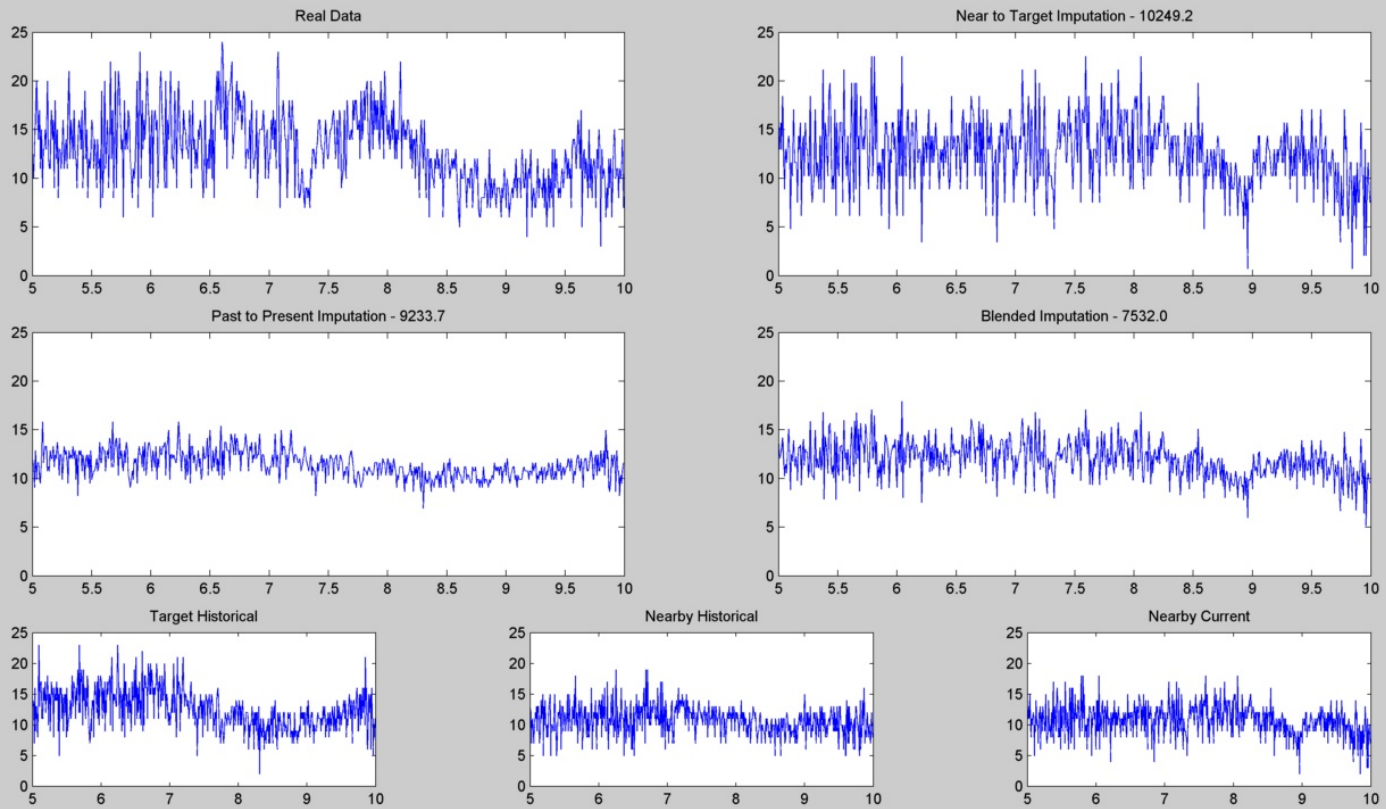


Speed - Correlation: 0.612

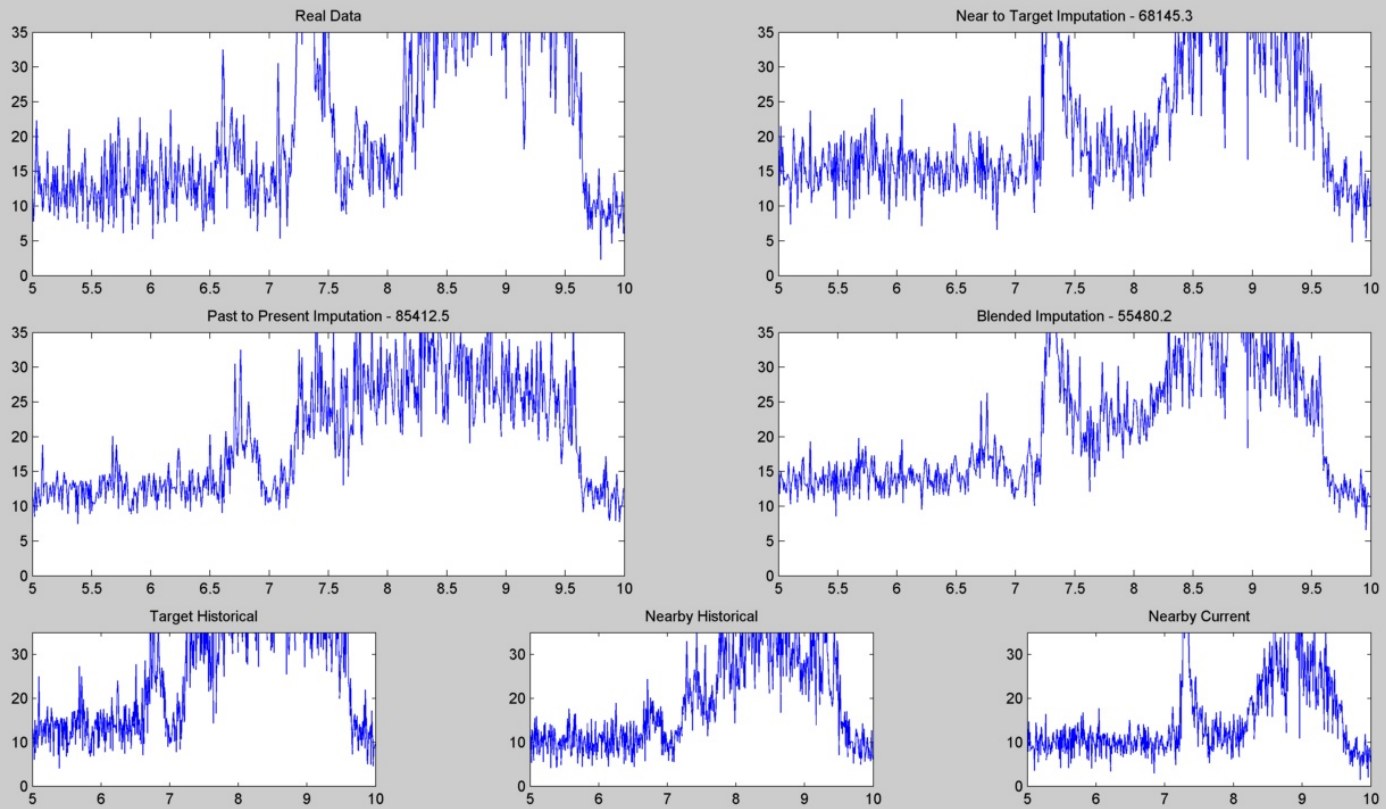


#	Target	Nearby	Current Date	Historical Date
14	1609	1610	10/13/2010	10/20/2010

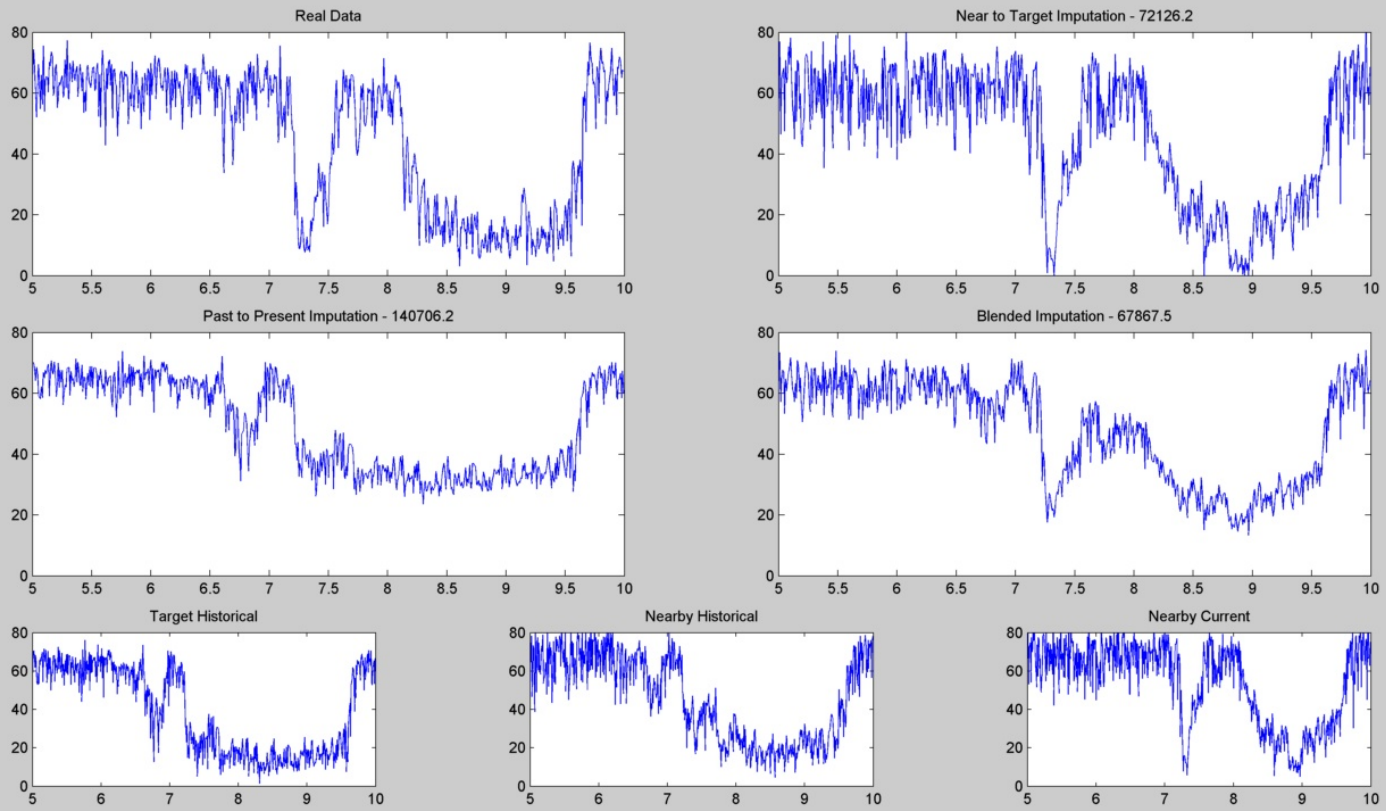
Volume - Correlation: 0.677



Occupancy - Correlation: 0.677

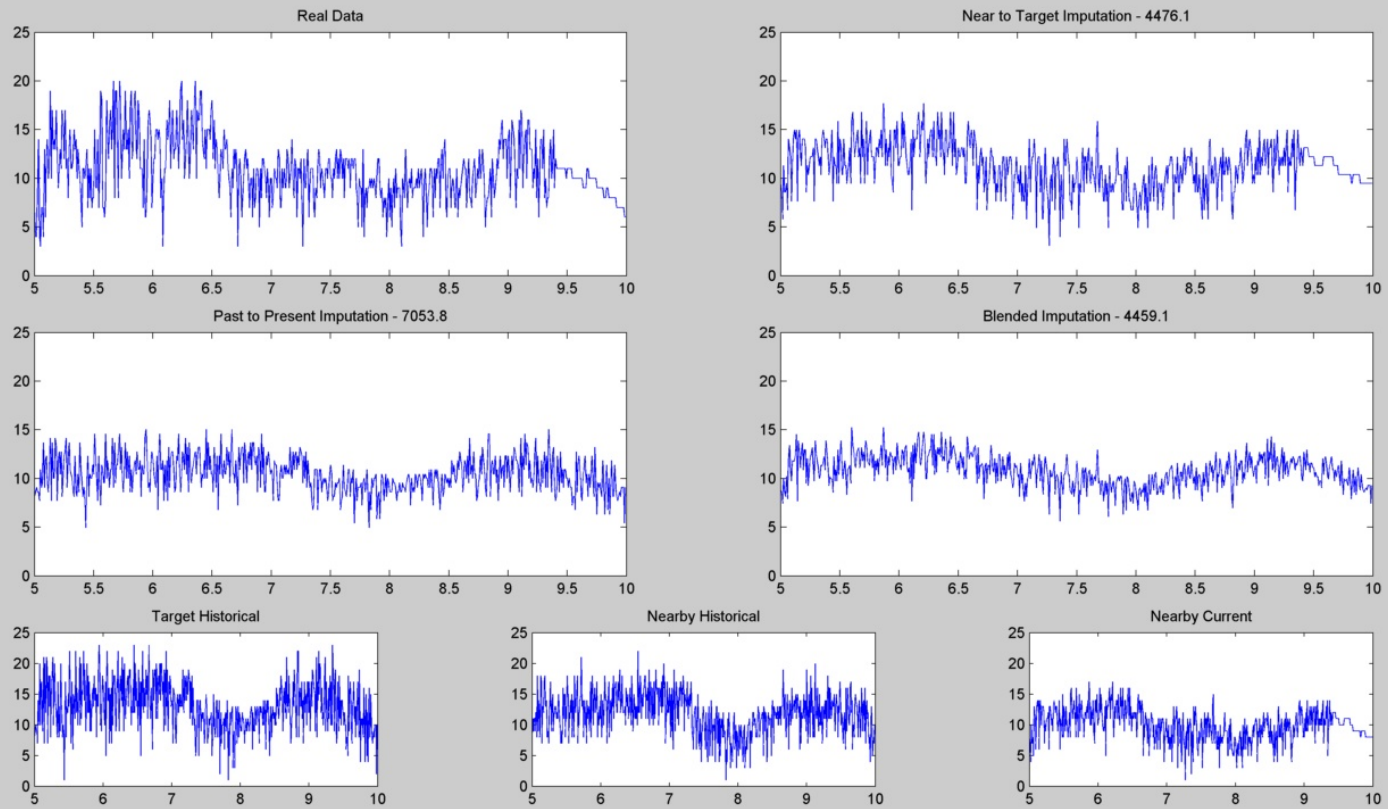


Speed - Correlation: 0.677

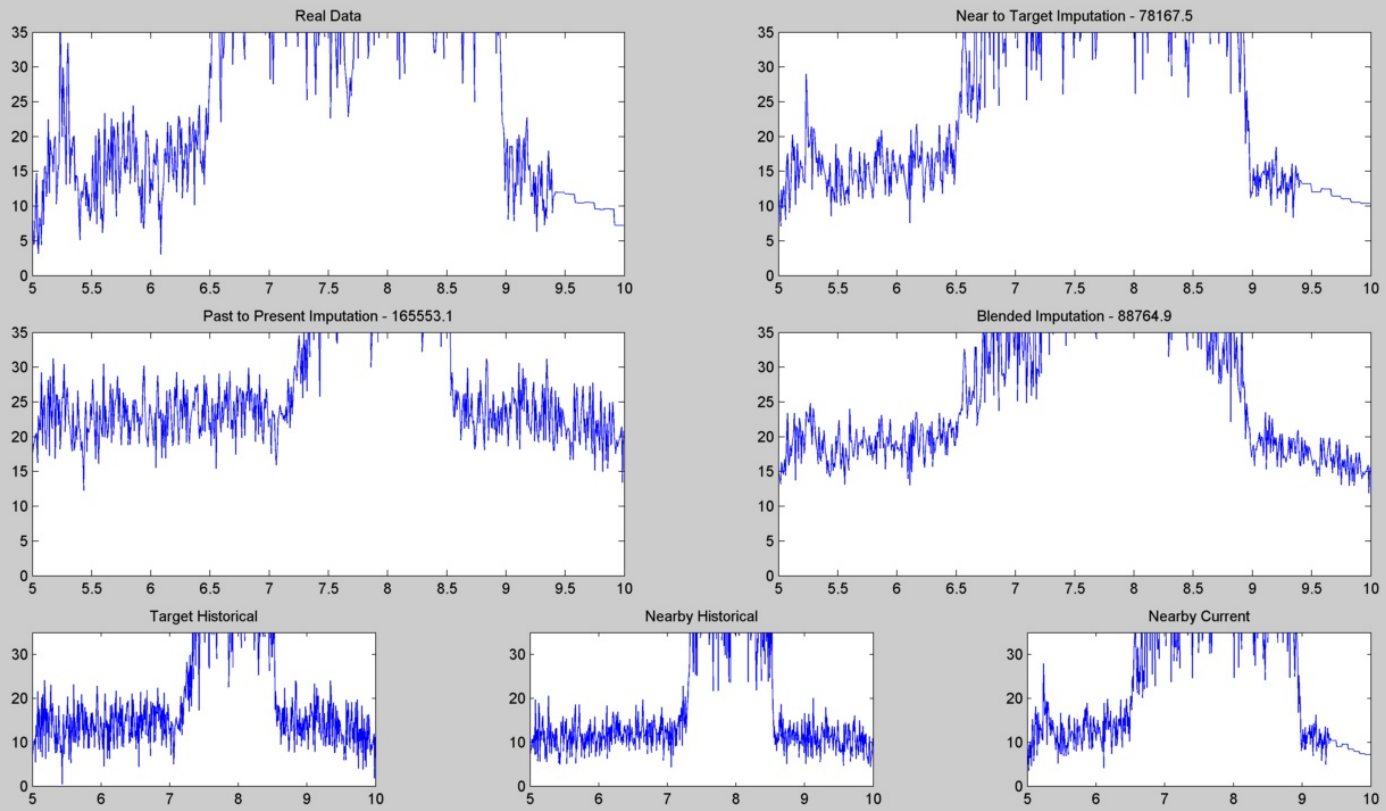


#	Target	Nearby	Current Date	Historical Date
15	2646	2645	10/26/2010	10/20/2010

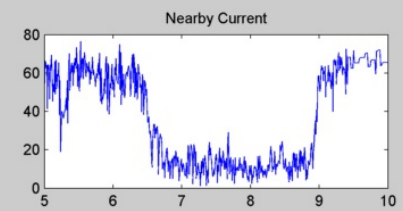
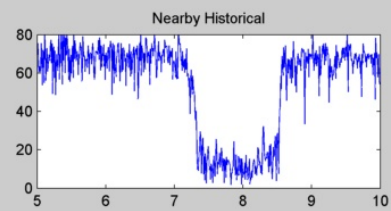
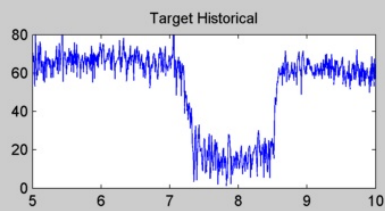
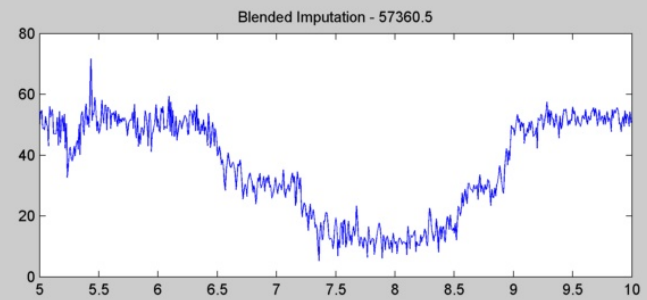
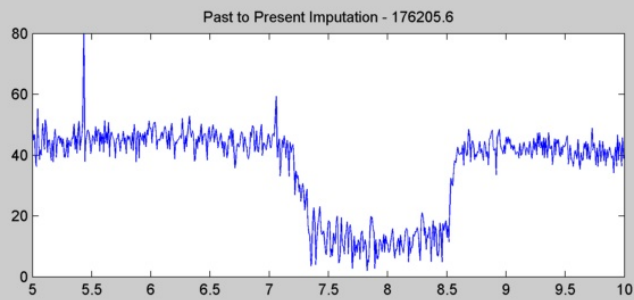
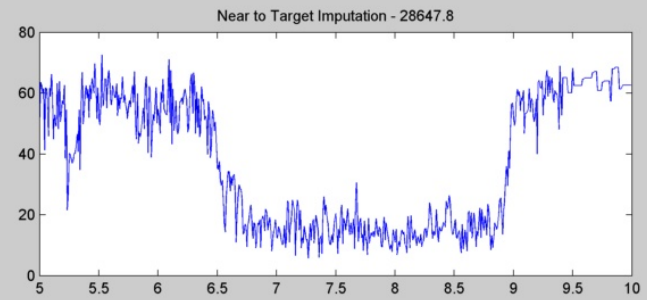
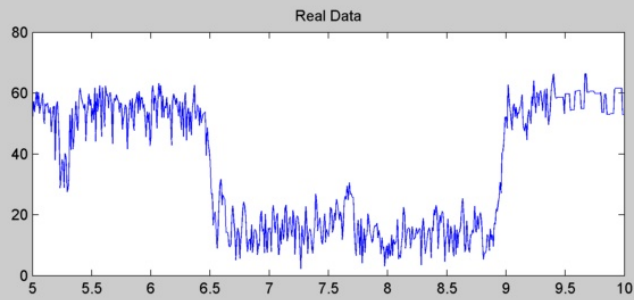
Volume - Correlation: 0.681



Occupancy - Correlation: 0.681

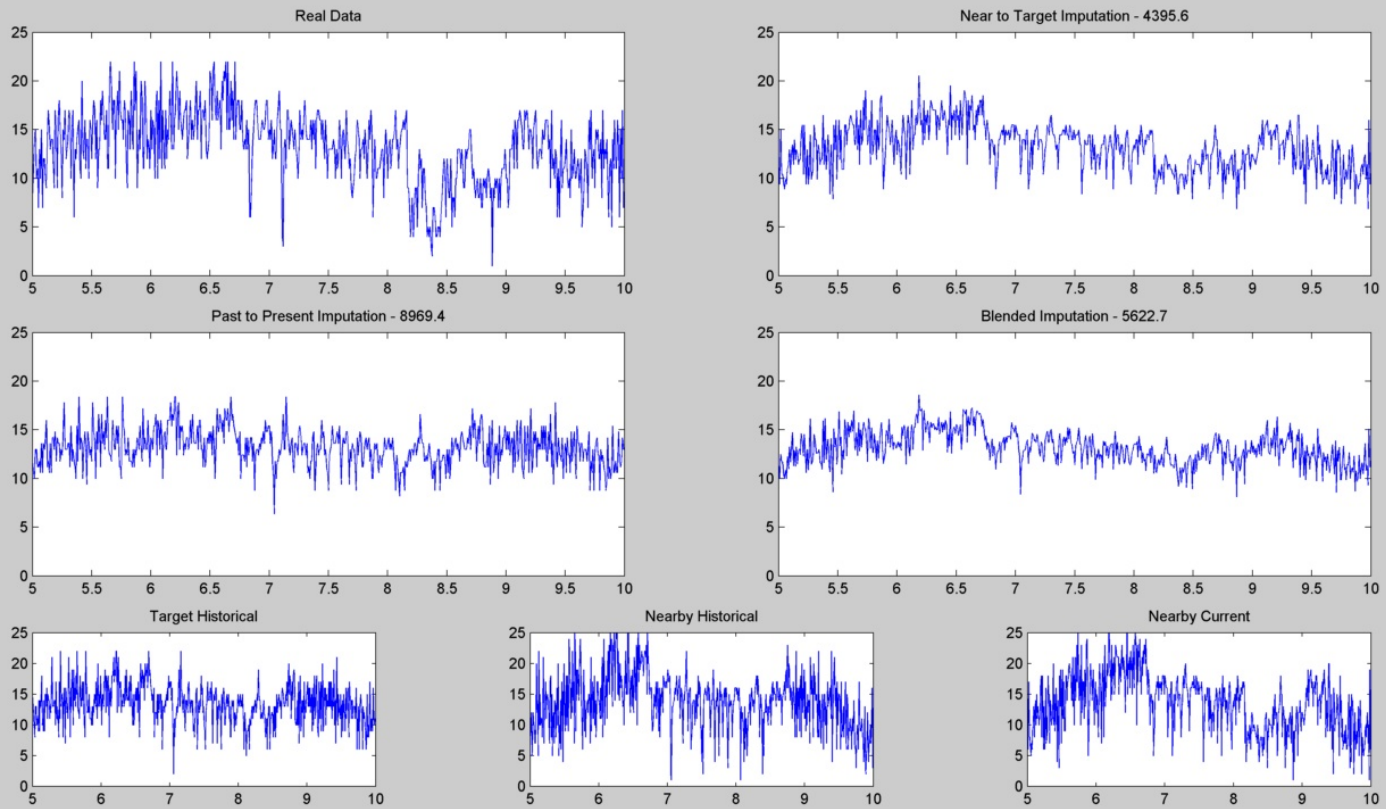


Speed - Correlation: 0.681

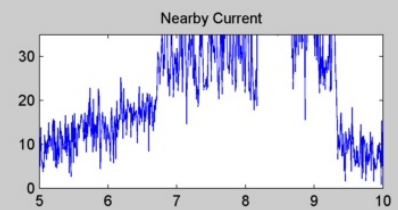
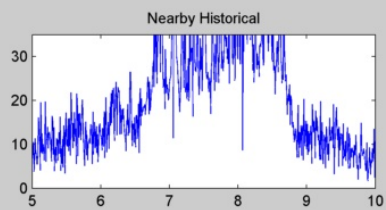
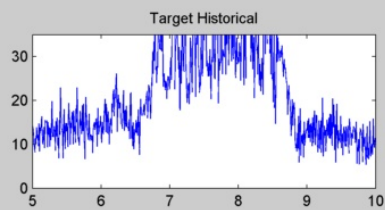
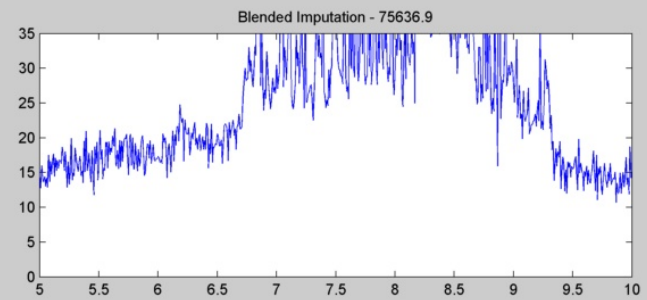
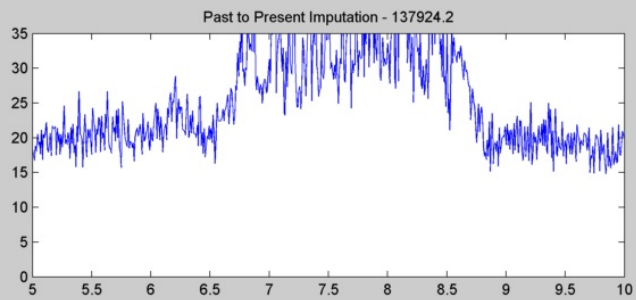
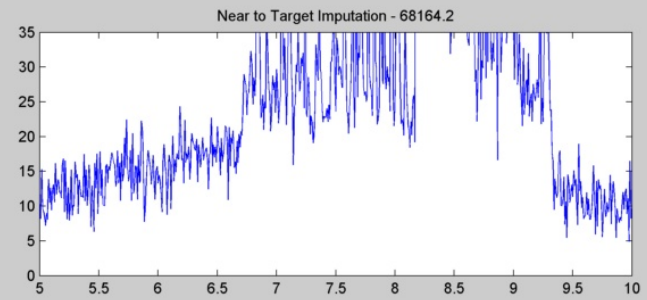
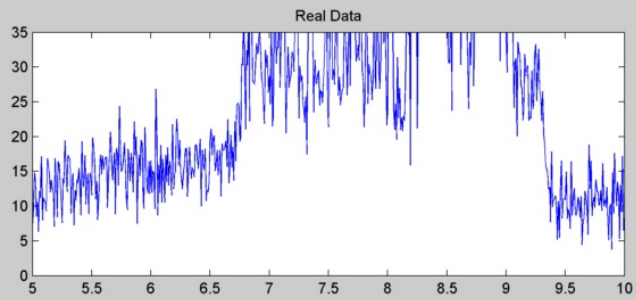


#	Target	Nearby	Current Date	Historical Date
16	2711	2712	10/28/2010	10/20/2010

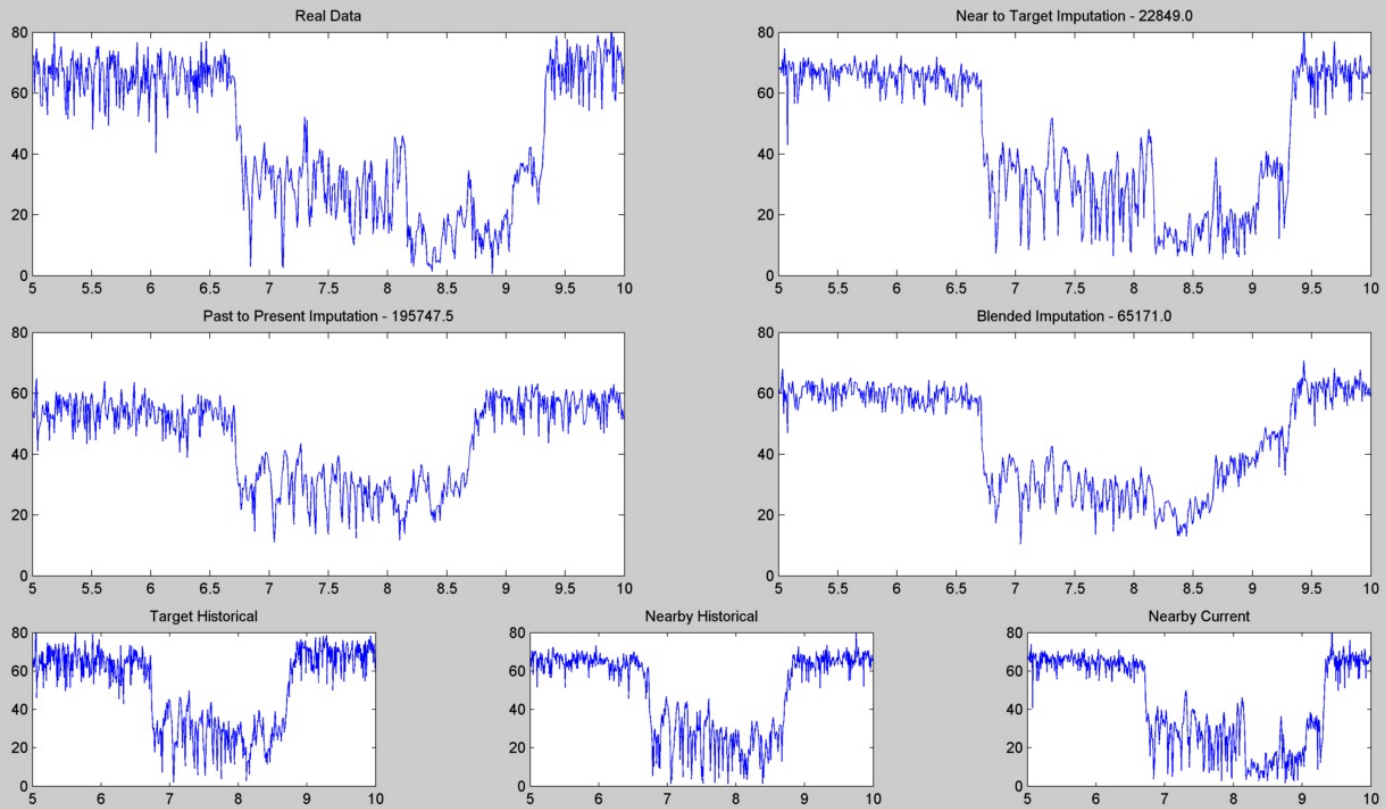
Volume - Correlation: 0.734



Occupancy - Correlation: 0.734

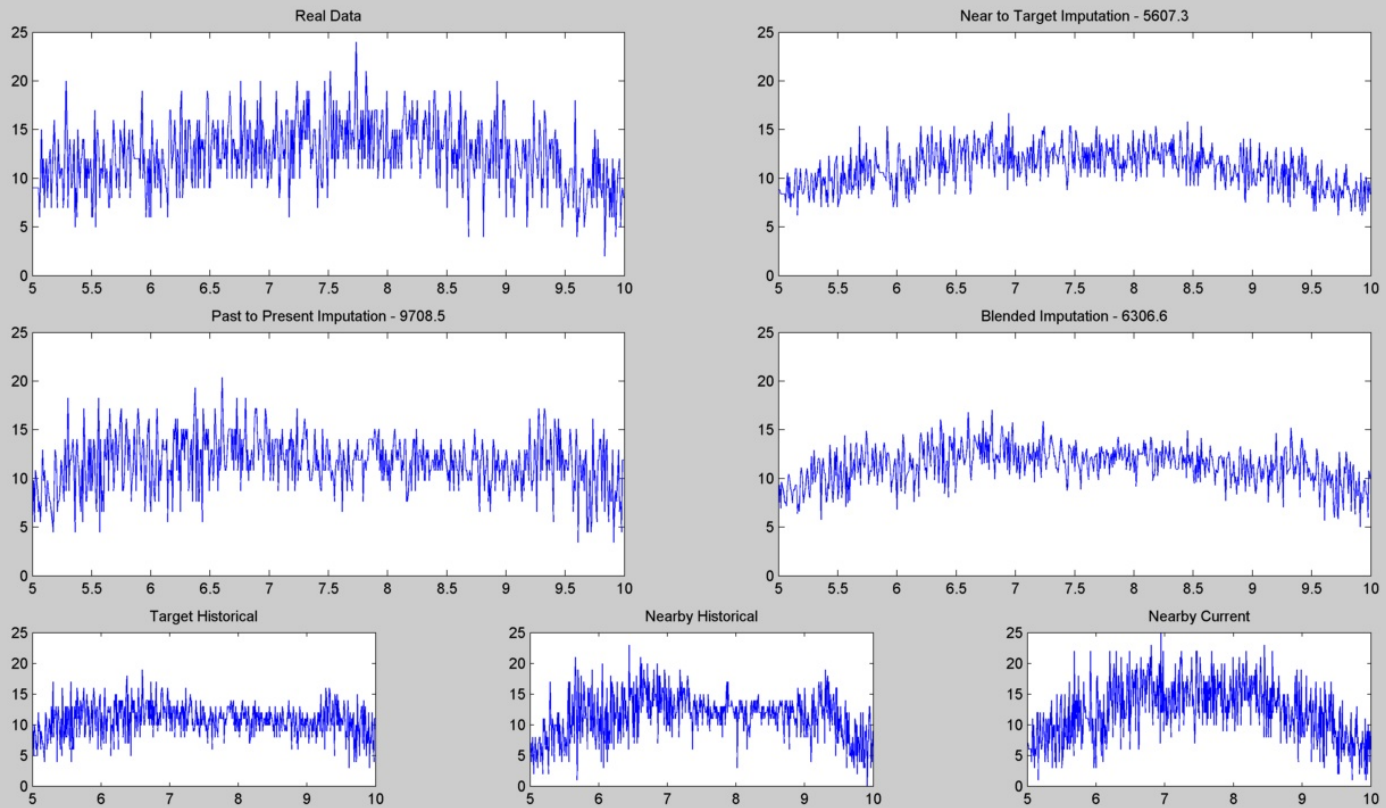


Speed - Correlation: 0.734

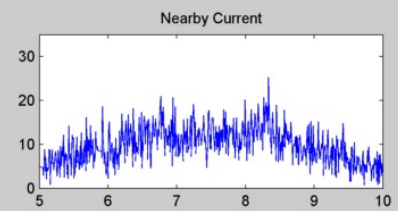
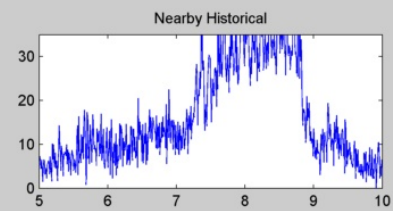
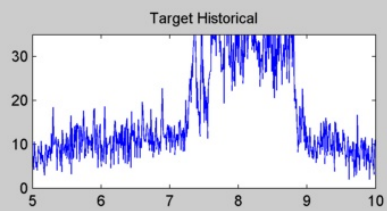
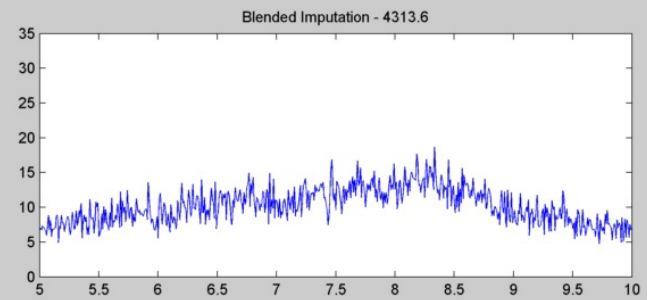
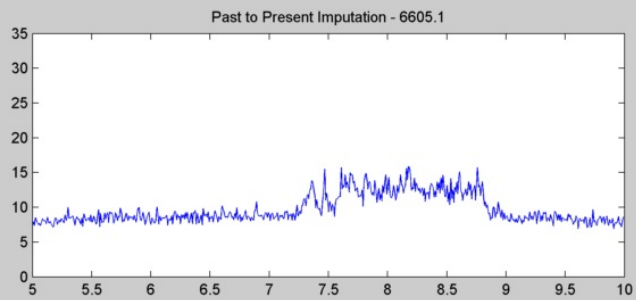
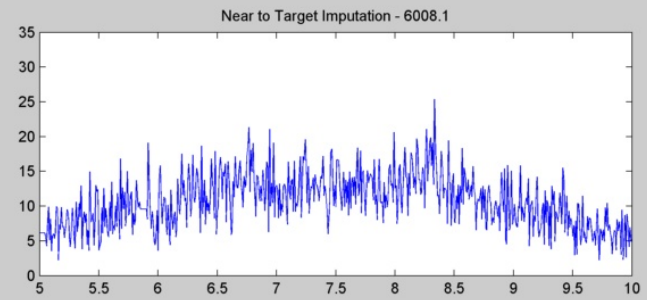
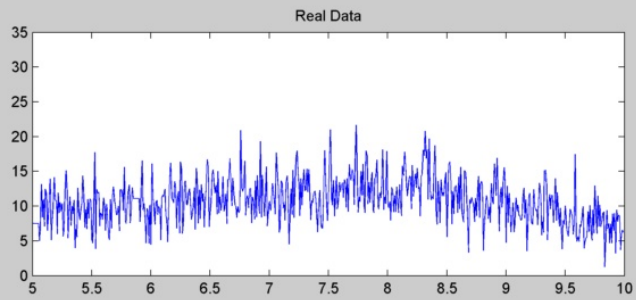


#	Target	Nearby	Current Date	Historical Date
17	4109	4110	10/27/2010	10/26/2010

Volume - Correlation: 0.687



Occupancy - Correlation: 0.687



Speed - Correlation: 0.687

