Data Visualization Tool for Monitoring Transit Operation and Performance

Abdullah Kurkcu
Department of Civil and Urban Engineering &
Center for Urban Science + Progress
Tandon School of Engineering
New York University (NYU)
New York, USA
ak4728@nyu.edu

Fabio Miranda
Department of Computer Science & Engineering
Tandon School of Engineering
New York University (NYU)
New York, USA
fmiranda@nyu.edu

Kaan Ozbay, Ph.D.
Connected Cities for Smart Mobility toward Accessible
and Resilient Transportation (C2SMART) Center
Department of Civil and Urban Engineering &
Center for Urban Science + Progress
Tandon School of Engineering
New York University (NYU), New York, USA
kaan.ozbay@nyu.edu

Claudio T. Silva, Ph.D.
Department of Computer Science & Engineering
Tandon School of Engineering
New York University (NYU)
New York, USA
csilva@nyu.edu

***Abstract*—Using the automated vehicle location data combined with other technologies such as automated incident reporting, transit decision makers can now execute a variety of real-time strategies and performance evaluations. In this study, we show that it is possible to develop an easy to use but powerful web-based tool which acquires, stores, processes, and visualizes bus trajectory data. The developed web-based tool makes it easy for the end users to access stored data and to query it without any delay or external help. Moreover, the tool allows the users to conduct a series of data visualization and analysis operations demonstrating the potential of a such web-based tool for future applications.**

Keywords—GPS bus data, visualization, web-based tools, performance measures, travel time, data analysis

I. Introduction

Travel times can be collected from a large number of potential data collection systems including fixed detectors such as inductive loops [1-3]. Such systems are experiencing a gradual but substantial change, moving away from being static and manually updated systems into highly integrated and interactive data-driven systems utilizing real-time data from smartphones and Global Positioning Systems (GPS) [4-7]. Transit agencies traditionally relied on manual data collection methods such as travel diaries and surveys to collect transit operation and planning data. Recent technological advances and extensive deployment of automated vehicle location (AVL) technologies make GPS data sources a promising and cost-effective way to monitor transportation systems. In New York City, most buses are operated by New York City Transit (NYCT) under the Metropolitan Transit Authority (MTA) serving a population of 15.3 million people [8]. The bus system includes 233 routes that serve the five boroughs of New York City with 5725 buses. Every bus in the NYCT's public transportation system is equipped with GPS devices. MTA's Bus Time Application Programming Interface (API) is an open-source development factory which provides real-time updates every 30 seconds on where the buses are located using very low-cost hardware and wireless communication technologies. It utilizes the Service Interface for Real Time Information (SIRI) standard developed by the European Committee for Standardization (CEN) [9].

The typical analysis of the public transit system involves creating a large display or map of the entire system containing numerical information for individual bus lines and stations toward meeting specific objectives. Measuring the performance of the system along with its sub-units allow decision makers to determine how well the agency is performing with respect to its service standards. Multi-domain smart city transportation systems require scalable, open platforms for the acquisition, processing, storing of Big Data with the ultimate goal of making individual and collective decisions both at the operational and planning levels. In this paper, we describe one such system, which provides data-driven analyses for investigating areas for improvement and evaluate existing transit systems using automatic bus location and status technologies. The information gathered from the developed tool can be utilized for online and offline performance analysis of transit operations using Geographical Information Systems (GIS) tools. This computerized tool allows users to combine existing data sources, increases productivity, and provides performance measures and up-to-date information to its users.

Combined with other technologies, such as automated incident reporting, it is now possible for transit managers to execute a variety of real-time strategies and performance evaluations. In this study, we show that it is possible to develop

978-1-5090-6484-7/17/$31.00 ©2017 IEEE

a simple yet powerful web-based tool to acquire, store, process and visualize bus trajectory data. This tool has an intuitive user interface that can be extended y incorporating functions that can improve the robustness of the tasks at hand. The fact that the tool is web-based makes it easy for the end users to access stored data and to query it without any delay or external help. Moreover, the tool enables the users to conduct a series of data visualization and analysis operations demonstrating the potential of such a web-based tool for future applications. The rest of this paper is structured as follows: previous work is summarized in the next section. Then, the open-source bus GPS data collection, cleaning and filtering approaches used for monitoring transit performance are described in detail. To understand and execute these transit performance statistics in an efficient and effective form, functionalities, the use interface, and sample visualizations of the developed web application are explained in the fourth section. The research findings and challenges in the development of building a web-based application for this study are concluded at the end of this paper.

II. Background

Measuring system performance is conceptually a significant task for transit agencies because this indicator is used to produce resource and asset allocation strategies into capital funds and operational enhancements. GIS-based performance analyses of transit systems have become popular in the recent years. Bertini and El-Geneidy [10] showed that transit performance measures could be generated to assess the functionality of the transit system by using the archived vehicle location data collected by transit providers. They pointed out that such data can provide insights about time-varying behavior which can be used to develop aggregate performance metrics. Berkow et al. [11] introduced a set of visualization tools indicating the reliability and variability of operating transit services. They combined different kinds of data sources to present performance measures not only statistically but also visually. With the help of technological advancements, higher resolution bus trajectory data become available to researchers. Glick et al. [12] investigated to identify traffic congestion along urban arterials using higher resolution bus location data. The results showed that bus trajectory data could be used to identify congestion, speed breakdowns, bus speed, and estimate queuing delays. Mesbah et al. [13] analyzed the performance of the Melbourne tram (streetcar) system through spatial and historical analysis of stop level automatic vehicle location data at a network-wide level. The analysis consisted of more than 165 thousand trips for 9 years. Their reliability and travel time analyses provided the critical targets for planners to improve the performance of the transit system as a whole. They concluded that the dwell time plays a major part of travel time and it should be minimized.

The majority of previous studies utilize commercial GIS tools to analyze transit performance measures for operations and planning [14]. One of the major drawbacks of using commercial software is the cost of purchasing the system and appropriate licenses for each user. Another limitation is that agencies have to provide spatial data similar to the format used by these tools. However, recent studies in the literature have started focusing on open architecture platforms which provide the more user-friendly environment and enhance the interoperability and scalability. For instance, Digital Roadway Interactive Visualization and Evaluation Network (DRIVE Net) is one such platform for data demonstration, archiving, analysis, and visualization that has been developed by Ma and Wang [14]. DRIVE Net platform utilizes automated fare collection (AFC) and AVL systems in Beijing and provides the network-level speed, route level travel time reliability, stop-level ridership, and headway variance. Similar to DRIVE Net, BusViz is also an open-source web-based application developed by Anwar et al. [15] which assists agencies and decision makers to better analyze and monitor large streams of field data by visualizing the performance of bus fleets. BusViz uses AFC data to survey bus stops to monitor bus arrivals [15]. It can also identify bus bunching by generating time-space diagrams of a certain bus route or a set of different bus routes to increase service capacity. SMARTBUS is another web-based system which has 3-layers for the evaluation and monitoring of the transit service [16]. The first layer estimates travel time and passenger demands fusing various data sources. The second layer analyzes the bus stop network to derive passenger patterns. Finally, the last layer delivers information about indicators to help users understand the impact of different factors on route-level bus delays. TaxiVis [17] is another example of an open-source system developed for the visualization of traffic data, but this time with a focus on taxi trips in NYC; they propose a new model that can interactively query over 500 million trips. The developed tool in this study can acquire, store, process and visualize not only historical transit data but also real-time information. In addition, more real-time data sources can be added to the tool because of its open-source nature.

III. Data

SIRI VehicleMonitoring Call [8] allows developers to request data about one, a set of, or all operating vehicles in the MTA Bus Time transit system. The responses from the MTA server to calls made by developers contain information including the ID of the vehicle, route name, direction, timestamp, and distance in miles, stop point name, expected arrival time, expected departure time, location, and bearing. The daily responses from the Bus Time API, totaling up to 3 terabytes, were stored at a local server for the whole year of 2015. All responses were successfully parsed and converted into one table containing only the required elements utilized by the tool. From the data, the Global Positioning System elements of the Bus Time data appear to be transformed to adhere to the shape of the reported bus route, eliminating the lateral variations from the street. The only elements reported by the vehicle itself are the time, location and head-sign. All the other elements are inferred by the Bus Time server which is prone to errors. These inferences have to be made in real-time by the server, to remove the chance to assess the validity of them and correct the errors. It has been observed that the Bus Time server sometimes makes errors in inferring the vehicle's trip ID and such inferred ID is not persistent for the same vehicle. Trip IDs play a fundamental role in the data analysis since much of the analysis requires grouping or sorting data by using these IDs.

Bus GPS data which is available through MTA API developer sources is stored in a web-based server. The system acquires latitude, longitude information from the BusTime feed as primary input. The system uses "vehicle id" and "distance along trip" fields to calculate average travel times and speeds for network links on bus routes. It also includes the capability to calculate point-to-point average travel times using GPS readings. One main performance measure using GPS data is the link-based travel times. The output is compiled only using adequately good GPS signals that are confirmed by a map-matching algorithm simultaneously. Raw dataset is not accessible and not to be altered by end-users. Although the developed system tries to visualize the whole year data, some days are entirely missing. The missing data may be caused by some uncontrollable factors such as weather; some mitigation plan may be devised to identify errors in the API system.

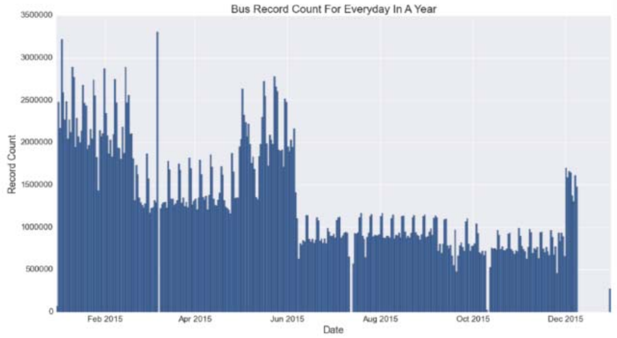

Fig. 1. Visualization of Records throughout 2015 - Source: Jiaxu Zhou [18]

Fig. 1 illustrates the irregularities in the data. The first data irregularity was in February 2015, second in May 2015, third in June 2015, and the last one in December 2015. It can also be seen that there is an extremely high number of records on March 7th, 2015 and March 8th, 2015 has missing data. One can infer that data for March 7th, and March 8th were merged, or the date reported in the responses is not accurate [18].

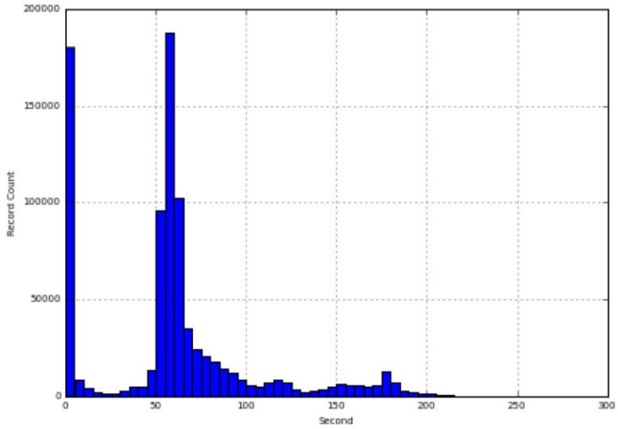

Fig. 2. The Frequency of Actual Intervals from Bus Time API - Source: Jiaxu Zhou [17]

Buses transmit their information every 30 seconds. Thus, the interval between Bus Time records is expected to be 30 seconds as long as the equipment works properly. However, the actual interval observed from the dataset collected turns out to be approximately 60 seconds [17]. In addition, this interval even gets longer when scheduled activities increase. In other words, if more buses are transmitting their location, the server slows down to respond to the API calls. Fig. 2 shows the frequency of actual intervals from the collected dataset. For example, the actual interval is zero when the system retrieves the same exact Bus Time record twice.

IV. The Assessment of User Needs

The research team held interviews with transit decision-makers to discuss desired functionalities for the web-based application. There were four main topics interested by the users which included the corridor definition and travel time queries; day, date range and time definition; output with different levels of aggregation, and the map output. The functionalities requested by front-end users are scored depending on their plausibility on a 1-5 scale, with 5 being most likely and 1 being least likely. After this evaluation, 41 potential functionalities are identified. Out of these 41 features, 25 are scored higher than 3 labeled as critical. Among these 25 critical features, 23 of them are identified to be implemented in the tool. TABLE 1 summarizes the requested functionalities. There were 2 functionalities that could not be implemented due to the open data lacking required information requested by the decision makers. For instance, the MTA Bus Time API data do not contain out of service or deadheading buses.

TABLE 1 Summary of Requested Functionalities

Plausibility	**# of Functionalities**	**Implemented**	**Not Implemented**
5 – (Highly Likely)	15	14	1
4 – (Likely)	10	8	2
Total	25	22	3

In addition to the requested features, the storage practice is optimized to use storage space effectively considering long-term data archiving for the users. The developed system has the capability to select desired time spans from historical data for the end users. It can create, and export output in comma-separated values (CSV), portable network graphics (PNG), GeoJson, or MS Excel spreadsheet format for later use. The system provides color-coded values in a legend for graphical results. It allows filtering the data by multiple bus lines and prompts error when there is no data. The tool can report the aggregated bus count data for selected segments within user specified time lengths.

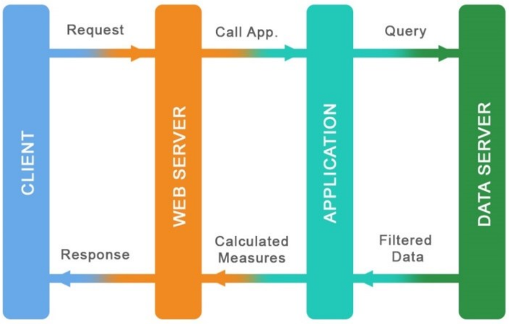

Fig. 3. System Architecture – Source: [19]

V. User Interface and Functionalities

This section introduces the features and design schemes of the front-end of the developed open-source web application that allows transit planners to focus on transit trends, identify problems, calculate travel time and stop level dwell time measurements. The architectural representation of the developed system can be seen in Fig. 3. Whenever users make any queries, the tool sends the required information to the web server. The server then calls the specific application such as data extraction, aggregation or transportation measures calculation.

Using the filters defined by the users, the application will query a MongoDB data server. This NoSQL database uses a document model, allowing for easy modification of the columns, and an intuitive integration with the application layer. On top of that, MongoDB also offers the same spatiotemporal queries that are provided by traditional databases, such as MySQL and PostGIS. After getting the requested results, the application will then calculate the measures and send them to the server to be presented to the user.

A. General Queries

The Bus Time data can be filtered by the time (hour and minute), day of week, month, year, direction, custom date range, bus ID, and bus line (single or multiple) using the developed web application. Fig. 4 below shows all the possible outcomes in the system. Two general information processing approaches are employed:

(1) Interactive and custom filters allow users to query and display information needed for the day-to-day and longer term performance monitoring and evaluation.

(2) Scenario comparison from static performance outputs generated by users which provide a snapshot of the transit system at multiple summary levels.

All the selection filters are explained in detail in the following sections.

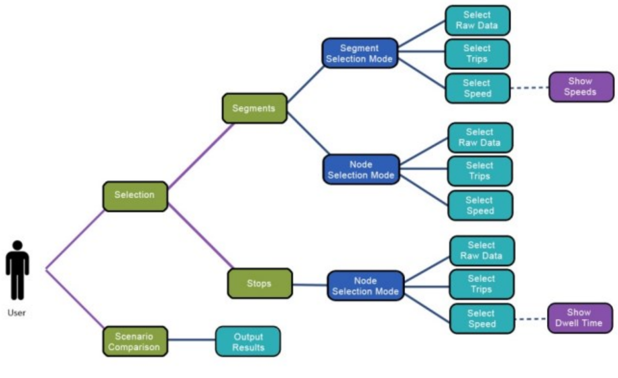

Fig. 4. User-Case View – Source: [19]

B. Spatial Selection Tool

It is possible to filter the Bus Time data by creating spatial filters on the map. The developed tool has three main modes that allow users to build a spatial filter and save it as a geoJSON file, an open standard format that describes geographical features and is supported by most GIS tools. Users can either define the start and the end point of the trips or create routes by clicking on consecutive segments.

1) Node Selection

In the node selection mode, the system will try to find matching trips between the first and the following clicks. Linear Integrated Ordered Network (LION) is a single line street base map representing streets and other linear geographic features in New York City. The light blue segments show the loaded LION layer which is by default from intersection to intersection. When the user clicks on the region of interest on the map, the system will snap the point to the nearest street. The direction of the clicks is not important since the travel direction can be selected using the existing filters in the system. The measures will be reported by using the data coming from all the matching trips between consecutively clicked points. An example selection can be seen in Fig. 5.

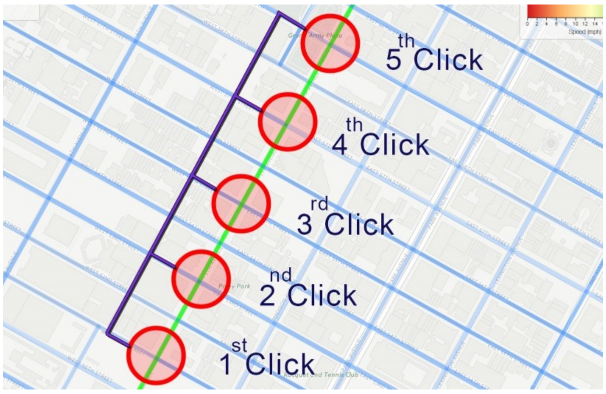

Fig. 5. Node Selection Tool – Source: [19]

2) Segment Selection

In the second selection method, the segment lengths are defined by the LION layer which allows users to click on the street segments to build routes. When users hover over the segment to be selected, the system will change the color to green to indicate the interested corridor. Once the segment is selected, its color will turn red. Although non-continuous segments can be selected using the tool, users are encouraged to click on every segment of the desired path. An example selection can be seen in Fig. 6.

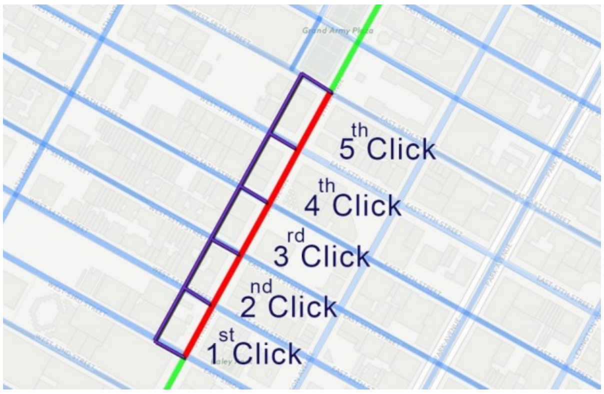

Fig. 6. Segment Selection Tool – Source: [19]

Selecting non-continuous segments is useful for comparing two different routes. After building the intended route on the map, users should save it as a geoJson file which can be used in the filtering process. The key difference between the two different selection methods is the way transportation measures are calculated.

3) Stop Selection

"Stops" option in the "Selection" tool will turn on the bus stops layer. This option will help users to be able to select individual bus stops using the third selection mode. Users can create reports for individual bus stops by the filters generated through this selection mode. The orange circles in Fig. 7 show the location of bus stops. The red circles show the selected stops.

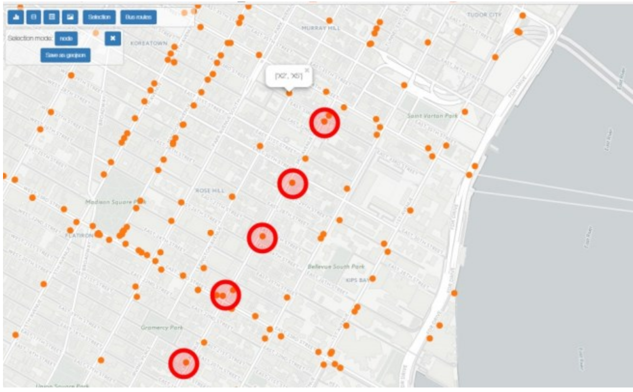

Fig. 7. Bus Stop Selection – Source: [19]

C. Travel Time Analysis and Visualization

If the spatial filter is created via the segment selection tool, the color-coded speed can be visualized on the map. When the user clicks on a segment, the system will show calculated speed values aggregated by bus line for the respective sections. As it can be seen in Fig. 8, the segments are color-coded according to the color scale on the top right of the screen. When a user clicks a segment, more information about that segment is shown in a pop-up.

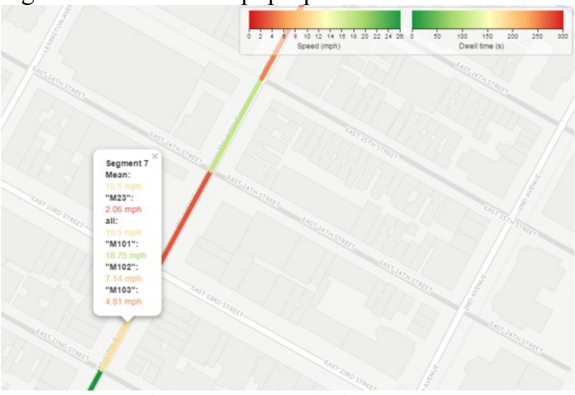
Segment 7
Mean:
10.5 mph
"M23":
2.06 mph
all:
10.5 mph
"M101":
18.75 mph
"M102":
7.14 mph
"M103":
4.81 mph

Fig. 8. Segment Selection Results

D. Dwell Time Analysis and Visualization

In the visualization tool, the system can visualize the dwell time on each stop if the stop selection tool is used. The nodes/stops will be color coded depending on the delay experienced by each element. When a user clicks a node, more information about that particular node is shown in a pop-up screen. The circles are color-coded according to the color scale on the top right of the screen. Fig. 9 shows an example of dwell time visualization.

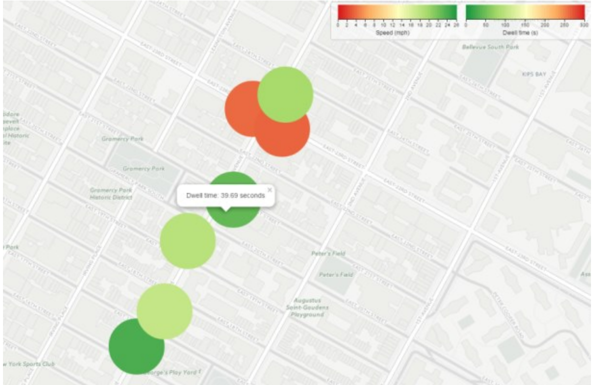
Dwell time: 39.69 seconds

Fig. 9. Dwell Time Analysis

E. Scenario Comparison

The scenario comparison tool will allow users to make before/after analysis for the desired segments. Users need to load the datasets containing exported information such as exported speed files using the tool. The system will automatically build the speed bar charts for all the bus lines for the respective segments. Users can also select individual bus lines by clicking "Line" button which shows all the available bus lines on the plot window. The bars in the bar chart are ordered according to the clicks made by the user when creating the segment selection. If the user hovers over a bar, then the corresponding segment will be highlighted on the map. An example use can be seen in Fig. 10.

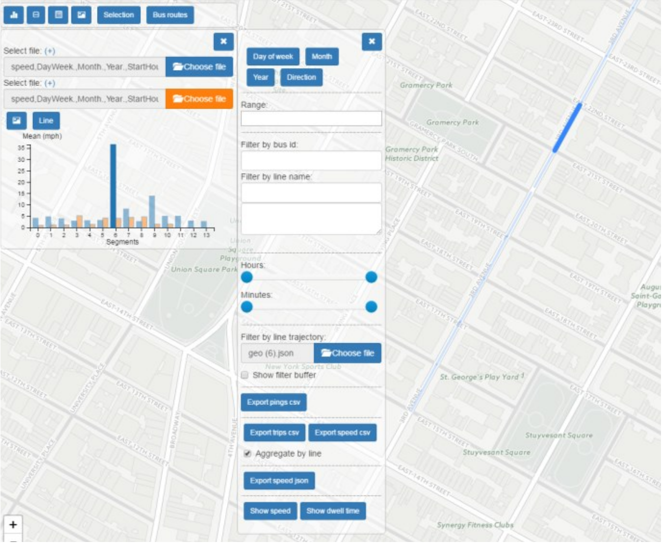
Select file: (+)
speed,DayWeek.,Month.,Year.,StartHou
Select file: (+)
speed,DayWeek.,Month.,Year.,StartHou
Filter by line trajectory:
geo (6).json
□ Show filter buffer
☑ Aggregate by line

Fig. 10. Scenario Comparison Tool

VI. Conclusions and Future Work

In this study, we showed that it is possible to develop a simple yet powerful web-based tool to acquire, store, process and visualize bus time data. This tool has an intuitive mapping user interface that can be improved by incorporating functions that can improve the robustness of the tasks at hand. The fact that the tool is web-based makes it easy for the end users to access stored data and to query it without any delay or external help. Moreover, the tool enables the users to conduct a series of data visualization and analysis operations demonstrating the potential of such a web-based tool for future applications.

One of the major challenges before and during the development of the tool was the estimation of the complexity of each task. After conducting interviews with decision makers, some tasks that were initially categorized as trivial turned out to be more complex, and so the time and effort dedicated to each task were constantly changing throughout the study. However, even though precious time was devoted to coding features that were ultimately scrapped or changed, this iterative process was fundamental in building an ultimately better tool. Another additional challenge was the fact that the users of the tool had different levels of familiarity with web-tools. Because of this, it was required to dedicate more time to make sure that all possible use scenarios were simple and intuitive. On top of that, certain constraints were added to the interface to make sure that the user did not deviate too much from the intended usage scenario. Software development always has the difficult task of balancing between scope, schedule, and resources. Creating an intuitive and easy-to-use tool that can satisfy the needs of users with different backgrounds is an enormous effort that goes beyond simply creating computational algorithms. It is about finding the sweet spot in a myriad of possibilities and constraints. The tool ultimately succeeds in this task and, more importantly, it provides an open and extensible platform that can be used and modified so that different metrics and processes can be tested as needed. One major improvement will be to incorporate other traffic data sources such as NYC taxi data and fixed traffic sensor data into this tool. Moreover, traffic information from commercial providers such as INRIX and WAZE can also be acquired and incorporated into the future versions.

VII. Acknowledgements

The contents of this paper only reflect views of the authors who are responsible for the facts and accuracy of the data and results presented herein. The contents of the paper do not necessarily reflect the official views or policies of sponsoring agencies. The work presented is partially funded by New York City Department of Transportation (NYCDOT), UTRC, UrbanMITS laboratory of the UrbanITS center at the NYU's Tandon School of Engineering.

References

[1] K. F. Petty, P. Bickel, M. Ostland, J. Rice, F. Schoenberg, J. Jiang, et al., "Accurate estimation of travel times from single-loop detectors", Transportation Research Part A: Policy and Practice, vol. 32, pp. 1-17, 1998.

[2] Z. Jia, C. Chen, B. Coifman, and P. Varaiya, "The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors", in Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, 2001, pp. 536-541.

[3] H. Guo and J. Jin, "Travel time estimation with correlation analysis of single loop detector data", Transportation Research Record: Journal of the Transportation Research Board, pp. 10-19, 2006.

[4] N. Uno, F. Kurauchi, H. Tamura, and Y. Iida, "Using bus probe data for analysis of travel time variability", Journal of Intelligent Transportation Systems, vol. 13, pp. 2-15, 2009.

[5] M. Rahmani, H. N. Koutsopoulos, and A. Ranganathan, "Requirements and potential of GPS-based floating car data for traffic management: Stockholm case study", in Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, 2010, pp. 730-735.

[6] E. F. Morgul, K. Ozbay, S. Iyer, and J. Holguin-Veras, "Commercial vehicle travel time estimation in urban networks using gps data from multiple sources", in Transportation Research Board 92nd Annual Meeting, 2013.

[7] E. Morgul, H. Yang, A. Kurkcu, K. Ozbay, B. Bartin, C. Kamga, et al., "Virtual Sensors: Web-Based Real-Time Data Collection Methodology for Transportation Operation Performance Analysis", Transportation Research Record: Journal of the Transportation Research Board, pp. 106-116, 2014.

[8] (February 15). The MTA Network. Available: http://web.mta.info/mta/network.htm

[9] (February 15). Service Interface for Real Time Information CEN/TS 15531 (prCEN/TS-OO278181). Available: http://user47094.vs.easily.co.uk/siri/

[10] R. Bertini and A. El-Geneidy, "Generating transit performance measures with archived data", Transportation Research Record: Journal of the Transportation Research Board, pp. 109-119, 2003.

[11] M. Berkow, A. El-Geneidy, R. Bertini, and D. Crout, "Beyond generating transit performance measures: visualizations and statistical analysis with historical data", Transportation Research Record: Journal of the Transportation Research Board, pp. 158-168, 2009.

[12] T. B. Glick, W. Feng, R. L. Bertini, and M. A. Figliozzi, "Exploring Applications of Second-Generation Archived Transit Data for Estimating Performance Measures and Arterial Travel Speeds", Transportation Research Record: Journal of the Transportation Research Board, pp. 44-53, 2015.

[13] M. Mesbah, G. Currie, C. Lennon, and T. Northcott, "Spatial and temporal visualization of transit operations performance data at a network level", Journal of Transport Geography, vol. 25, pp. 15-26, 2012.

[14] X. Ma and Y. Wang, "Development of a data-driven platform for transit performance measures using smart card and GPS data", Journal of Transportation Engineering, vol. 140, p. 04014063, 2014.

[15] A. Anwar, A. Odoni, and N. Toh, "BusViz: Big Data for Bus Fleets", Transportation Research Record: Journal of the Transportation Research Board, pp. 102-109, 2016.

[16] S. Ram, Y. Wang, F. Currim, F. Dong, E. Dantas, and L. A. Sabóia, "SMARTBUS: A Web Application for Smart Urban Mobility and Transportation", in Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 363-368.

[17] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips", IEEE Transactions on Visualization and Computer Graphics, vol. 19, pp. 2149-2158, 2013.

[18] Y. C. Jiaxu Zhou, Matthew Urbanek, Bonan Yuan, Sara Arango-Franco, Huy T. Vo, Kaan Ozbay, "Bus Reliability Metrics Using Public MTA Bus Time Data", 2016.

[19] "Develop Data Storage and Access Platform for MTA Bus Time Data", New York City Department of Transportation, New York January, 2017