A Scalable Data Analytics and Visualization System for City-wide Traffic Signal Data-sets

Dhruv Mahajan, Yashaswi Karnati, Tania Banerjee, Varun Reddy Regalla, Rohit Reddy, Anand Rangarajan and Sanjay Ranka

Abstract— **The advent of new traffic data collection tools such as high-resolution signalized intersection controller logs opens up a new space of possibilities for traffic management. In this work, we describe the high resolution datasets, apply appropriate machine learning methods to obtain relevant information from the said datasets and develop visualization tools to provide traffic engineers with suitable interfaces, thereby enabling new insights into traffic signal performance management. The eventual goal of this study is to enable automated analysis and help create operational performance measures for signalized intersections while aiding traffic administrators in their quest to design 21st century signal policies.**

I. Introduction

Mitigating traffic congestion and improving safety are the important cornerstones of transportation for smart cities. An INRIX study[1]found that in 2017, traffic congestion cost nearly \$305 billion and caused Americans to lose 97 hours per person in gridlock. This costs the U.S. \$87 billion annually in lost time. Many drivers are frustrated due to long (but potentially preventable) delays at intersections. The Governors Highway Safety Administration (GHSA) found[2] that pedestrian deaths have steadily increased with the 6590 deaths in 2019 an estimated 60 percent increase over 2009. Addressing these challenges requires a thorough understanding of traffic patterns not only at intersections but on the streets and in the overall network. Current performance evaluations include a limited comparison of before-and-after travel-time data to demonstrate the effectiveness of signal retiming efforts. However, traffic patterns vary dynamically during a day as well as globally within the network, and there is a need for continuous monitoring and evaluation of signal timing parameters based on performance and fluctuation demands.

The data available at each intersection can be broadly divided into signal timing data and vehicle detector data (gathered from loop detectors). The former consists of traffic movement and timing information for different phases, while the latter consists of arrival/departure and occupancy information for vehicles. In the past, this information was available at coarse levels of granularity (for example, traffic movement counts by the hour) limiting their use and the ability to discover cycle-by-cycle changes. Methods such as the Purdue Coordination Diagrams [1], [2] have been shown to be useful in understanding signal behavior and potential bottlenecks using signalized intersection datasets.

The availability of high resolution (10 Hz) controller logs opens a broader range of possibilities that were not available in previous systems. Additionally, this data, in many cases, is available with small latency (a few minutes) making it amenable to real-time decision making and addressing of bottlenecks. However, this plethora of information without proper decision-making tools adds a burden to transportation professionals. Many of them evaluate this information using ATSPM[3] tools and analyze the collected information one intersection at time. This is challenging even for small cities comprising of a few hundred traffic intersections. There is a need for a system that provides corridor-level and city-level information in a succinct and actionable form. In this paper, we describe a system that partially addresses this need. Our system leverages machine learning methodologies for data collected from a large number of intersections to derive key spatio-temporal traffic patterns in a city and then interactively allows a traffic engineer to focus on key challenges or improvements that can be carried out to alleviate them. Additionally, our system provides an analysis of traffic interruptions by observing changes in traffic at detectors, approaches, and intersections. The key modules of the system are now described.

1) Ranking: Several measures of effectiveness (MOEs) are used in the discipline [1], [3]. Our systems allow the user to rank or select intersections based on split failures and ratios of arrivals on red vs. green. Additionally, using a combination of these two metrics, we subdivide the intersections into several categories.
2) Clustering: Intersections with similar behavior or performance are grouped together using machine learning techniques. This approach is particularly useful when dealing with a large number of intersections and is carried out along both space and time. The system discovers and highlights signals on a corridor that preform similarly.
3) Change detection: We have developed a change detection algorithm that can detect statistically significant changes at an intersection level as compared to previous, similar time periods. This approach can be used to determine unexpected behavior or change in traffic

The authors are affiliated with the Dept. of Computer and Information Science and Engineering, Univ. of Florida, Gainesville, FL {dhruvmhjn,yashaswikarnati,tmishra}@ufl.edu, {vreddy.regalla,rkessireddy}@ufl.edu, {anandr,sranka}@ufl.edu

[1] https://inrix.com/scorecard/

[2] https://www.ghsa.org/resources/news-releases/pedestrians20

[3] https://udottraffic.utah.gov/atspm/

patterns.

4) Incident detection: Using time series analysis, we derive extended time periods of significant traffic reduction for a detector or for an approach. A spatio-temporal presentation of this information is useful to derive key areas of traffic interruptions.

We have implemented this system, which can be executed in parallel on a multicore machine and can handle data from thousands of intersections. The system can process six months of data for 300+ intersections (roughly 1 Terabyte) in less than 6 hours using a 50-core processor. A visualization module allows the user to select spatial and temporal regions of interest in an interactive fashion.

The rest of the paper is organized as follows. In Section II, we first describe the pre processing steps that are performed to convert raw controller logs to measures of effectiveness (MoEs). In Section III, we introduce the key modules in the system. These include: intersection ranking and categorization, clustering methods to highlight spatio-temporal patterns in the performance of intersections, change detection and incident detection. In the fourth section, we detail our visualization framework. In the final two sections, we detail the overall workflow, present results to demonstrate the performance & scalability of the system and summarize our contributions. We also discuss how our techniques can be incorporated into existing ATSPM systems.

II. Data-set

The availability of high resolution (10 Hz) controller logs opens has open up a broad range of possibilities in terms of traffic intersection monitoring and performance metrics. In the following, we describe the key steps required for information processing.

A. Intersection Controller Logs

Traffic signals are crucial in managing vehicular and pedestrian traffic at an intersection where two or more road segments meet. The new generation of signal controllers, based on the latest Advanced Transportation Controller (ATC) [4] standards, are capable of recording signal events as well as vehicle arrival and departure events at a high data rate (10Hz). This allows us to compute signal performance metrics such as arrivals on red, arrivals on green, and platooning ratios on a cycle-by-cycle basis [3].

B. Data Collection

We can ingest data on a daily basis for the advanced controllers of type NTCIP 76.x, ATC. The detailed data collection process is described as follows. Each controller stores 24 hours of this data. This data is collected once a day using FTP by providing the IP addresses, which are local addresses to the remote network. A script can be used to initiate a FTP connection to each controller, downloads the stored data, decode the data to ASCII format, and upload the data to a local computer for further processing. The raw data is then processed, and the required information extracted and stored to a database.

C. Detector Configuration or Detector to Channel Mapping

Meaningful interpretations of this high resolution data lead to the computation of useful performance measures at intersections. For this, certain secondary sets of data or information is needed. The most *critical* requirement for interpretation of the high resolution signal event data logged in a controller is the detector mapping information. To compute performance measures or measures of effectiveness (MOE) from the controller logs, we need to have detector-to-phase mappings. These mappings indicate the location of a detector, such as advanced or stop-bar, and the phase in which it detects vehicles. In many practical situations, these mappings are missing (e.g., the infrastructure was built decades ago, and the mappings are not available in a machine-readable form) or incorrect (e.g., during maintenance or addition of new lanes, the contractor forgot to update the mappings).

The lack of a specific location for a detector limits automated systems (like [5] and others) as they are unable to compute the performance measures that depend on the vehicular arrivals and departures for a particular detector or direction of movement.

The goal of this module is to find the best mapping of detector channels to phases and to classify detectors as stop bar detectors or advanced detectors based on events in the high resolution controller logs. These events include a change in the signaling state (for example, green, yellow, or red for vehicles, and walk, flashing do not walk, and do not walk for pedestrians) and a change in the detector state (based on whether the detection area is occupied or not). Our machine learning-based algorithms are driven by the intuition that the traffic on a detector during the green phase will be higher than that of the corresponding red phase [4].

D. Data Volume and Velocity

The total amount of data collected from each signal per day is between 50 to 100 MB (in ASCII format), and about a third of this is in the native binary format. Since the data is collected directly from the controllers, it has a high degree of veracity. The daily data download, decoding, and upload for each intersection require less than a minute.

III. Functional Architecture & Key Modules

The overall approach seeks to process data and aggregate it at cycle-by-cycle level. It is worth noting that cycle times, in general, are variable throughout the day for each intersection. This cycle level data is then used to generate several measures of effectiveness that are further aggregated to fixed size intervals (e.g., 15 minutes or an hour). Figure 1 provides the key modules and the overall workflow. In the rest of the section, we summarize each of the modules.

[4] https://www.ite.org/technical-resources/standards/atc-controller/

[5] https://udottraffic.utah.gov/atspm/

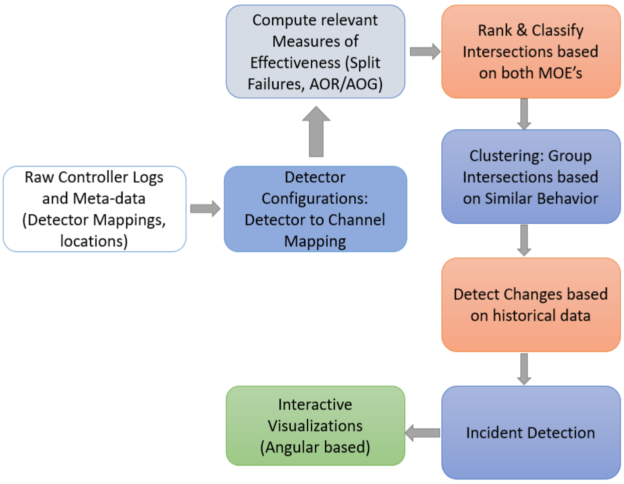

Compute relevant Measures of Effectiveness (Split Failures, AOR/AOG)
Rank & Classify Intersections based on both MOE's
Raw Controller Logs and Meta-data (Detector Mappings, locations)
Detector Configurations: Detector to Channel Mapping
Clustering: Group Intersections based on Similar Behavior
Detect Changes based on historical data
Interactive Visualizations (Angular based)
Incident Detection

Fig. 1. The figure describes the overall workflow of the system. Starting with raw data as input, we follow these steps to get to the interactive dashboards presented in IV.

A. Ranking and Classification

We currently use demand-based split failures (computing red occupancy and green occupancy ratios) and the ratio of arrivals on red to arrivals on green (AoR/AoG) as measures of effectiveness (MOEs) of an intersection. These measures serve as good proxies for the level of traffic demand and effectiveness of signal timing, respectively. The possibility of using other measures is being explored further.

Once these measures are computed, they can be used for both filtering (using a threshold) or ranking. This allows traffic engineers to focus their effort on the most problematic intersections. A combination of the above MOEs is them used to categorize intersections into four broad classes:

1) Low split failures, Low AoR/AoG: Well timed and utilized intersections
2) Low split failures, High AoR/AoG: Low demand but potential for timing improvements
3) High split failures, Low AoR/AoG: Potential capacity problems
4) High split failures, High AoR/AoG: High demand and potential for timing optimizations

Additionally, intersections with detection issues or missing data can be derived if these measures are very high or very low for extended periods of time.

B. Clustering

Daily data for each intersection based on MOEs is represented as a vector. The length of the vector is based on the number of intervals into which the entire day is divided. For example, if the data is aggregated at an hourly basis, the length will be 24. A weighted graph is first constructed for all intersection and day pairs based on distances between the vectors. Nonlinear dimensionality reduction, followed by clustering in the reduced space is then used to produce the clusters of similar behavior based on the MOEs. For more details the reader is referred to [5].

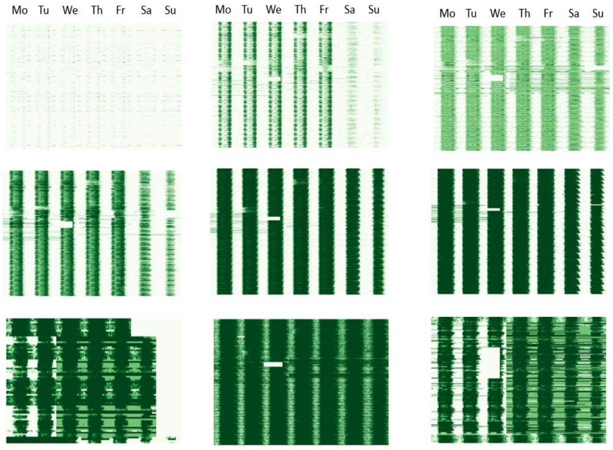


Fig. 2. The nine clusters corresponding to distinct demand patterns discovered in the data. Each plot represents one cluster, and reach row represents the weekly behavior of an intersection that belongs to the cluster. We order these clusters from low demand to very high demand.

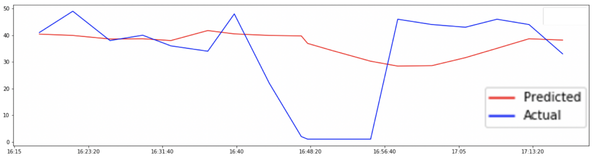


Fig. 3. This is an example of a major interruption. Note the significant deviation of traffic volumes from predicted volumes(amount and duration).

A representative set of cluster centers is derived and described in Figure 2. Sometimes the intersections belonging to a cluster are spread over geographic regions several miles apart. While these intersections may be performing similarly, there is limited real value in having such distant intersections in the same cluster. A second round of processing is performed to split a cluster of intersections into multiple disjoint clusters based on spatial or corridor locality. A geographical indicator such as primary road names or distance between the intersections is used for this purpose.

C. Change Detection

This module can be used to discover significant changes in signal performance. Our methods detect temporal changes in signal performance, and/or detect periods with changes in many signals, and automatically highlight the change or evolution of intersection performance with time. The following approach is used:

- A significant change in the performance of the intersection over time can be detected by observing the evolution of the intersection's cluster membership over time.
- A change in the lower dimensional projection of the data representing the intersection performance can be used as a change detection measure.

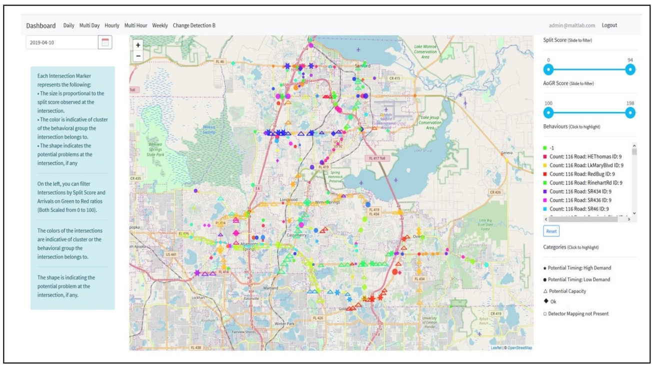
In practice, the two methods are combined into one overarching method. Details are provided in [6].

D. Interruption Detection

Managing traffic interruptions is one of the crucial activities for any traffic management center. These interruptions may be caused by traffic accidents, vehicle breakdowns, debris, etc. Further, this should be done in (near) real-time so that proactive actions can be undertaken for mitigation. Broadly, we define an interruption to be any time period where the amount of traffic is significantly lower than normal or predicted traffic for a significant period of time. A *large* traffic interruption is defined on the bases of two parameters (see Figure 3):

1) The magnitude of deviation (percentage reduction) of observed traffic volumes from predicted volumes. This is measured in terms of the percentage dip of the actual traffic volume vs. the predicted value. Common sense dictates that the greater the deviation, the larger the interruption.
2) The duration (in seconds) for which the actual traffic volume is less than a *baseline* predicted volume. Again, a long duration heralds a large interruption.

The reader is referred to [7] for further details.


Dashboard
Daily
Multi Day
Hourly
Multi Hour
Weekly
Change Detection B
2019-04-10
admin@maltlab.com
Logout
Split Score (Slide to filter)
0
94
AoGR Score (Slide to filter)
100
198
Behaviours (Click to highlight)
Reset
Categories (Click to highlight)

Each Intersection Marker represents the following:
- The size is proportional to the split score observed at the intersection.
- The color is indicative of cluster of the behavioral group the intersection belongs to.
- The shape indicates the potential problems at the intersection, if any

On the left, you can filter Intersections by Split Score and Arrivals on Green to Red ratios (Both Scaled from 0 to 100).

The colors of the intersections are indicative of cluster or the behavioral group the intersection belongs to.

The shape is indicating the potential problem at the intersection, if any.

🟩 -1
🟥 Count: 116 Road: HEThomas ID: 9
🟨 Count: 116 Road: LkMaryBlvd ID: 9
🟥 Count: 116 Road: RedBug ID: 9
🟩 Count: 116 Road: RinehartRd ID: 9
🟪 Count: 116 Road: SR434 ID: 9
🟪 Count: 116 Road: SR436 ID: 9
🟦 Count: 116 Road: SR46 ID: 9

★ Potential Timing: High Demand
● Potential Timing: Low Demand
△ Potential Capacity
◆ Ok
□ Detector Mapping not Present

Fig. 4. Dashboard showing the clustering and classification results for a single day. The results show that intersections on the same corridor demonstrate similar behavior throughout the day.

IV. Visualization

We have developed a visualization module that allows the user to derive trends and hotspots in their city using the modules described in the previous section. The key features of this module are the following:

1) The user can select a small subset of intersections based on MOEs of interest. This is important as it allows the user to focus on problem areas. Each signal is represented by an icon based on the four categories derived by the combination of the two MOEs. The size of the icons, used to represent each of the different categories (based on the two MOEs), is proportional to the relative severity. This allows the user to visually compare the different intersections.
2) The user can hover on a particular intersection to get a detailed description of the MOEs and other relevant information at a granular level. Examples of such information include the signal ID, the number of split failures that happened on an hourly basis for the major approaches, the number of arrivals on red and green and the number of pedestrian actuations.
3) The intersections in a cluster with similar behavior can be easily identified because they have the same color. This allows the user to observe spatially and temporally similar behavior across multiple hours or days. Further, the user can highlight all the intersections represented by a particular cluster. For each cluster, the behavior legend presents the corresponding color, the number of members, the name of the road where the members may be found, and the days of the week that the cluster was observed.
4) There are different screens for each functionality. Further, different screens allow the user to access information for a single time period or multiple time periods. The former is useful in looking at details for a single time period while the latter allows the user to see comparisons between time periods or trends. The time periods for multiple period screens can be chosen by a drop-down menu, and the user can flexibly chose two to ten time periods.

Figure 4 shows the results on a single day. Many signals on the same corridor get grouped together, showing that they performed similarly during the day. The clustering results of a multiday dashboard in Figure 5 show that for many intersections, the performance is similar during weekdays but differs from the weekends. For this particular week, many of the intersections performed well on Sunday but had potential capacity issues on weekdays.

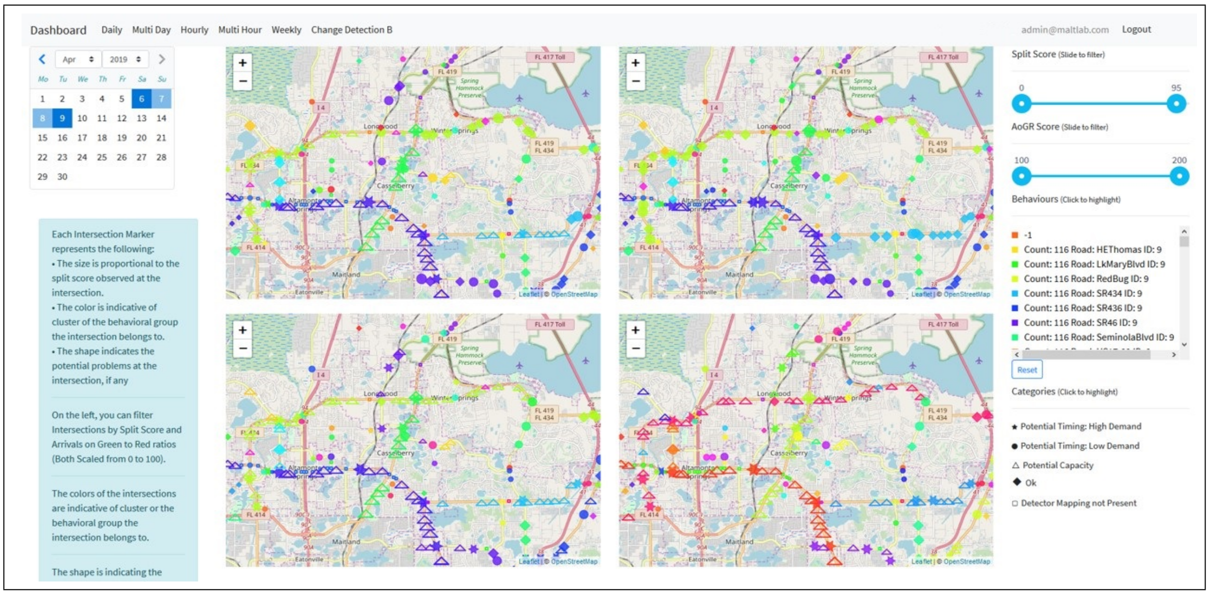
The clustering technique is sensitive enough to separate intersections with granular differences between the observed behavior that are limited to only a few days. i.e. certain behavior can occur only on weekends whereas other patterns exist throughout the weekdays. Similar screens are available for performing this analysis on an hourly basis . Thus, the visualization system can be used to understand key behaviors in a grid or network of signalized intersections. It can be used to understand the hours or days for which the traffic patterns are similar and the time periods for which there might be some problems

Figure 6 shows the change detection screen of the dashboard. The user can select two time periods for comparison and the system provided differences in behavior. If only one time period is chosen, the system automatically chooses a baseline (e.g., for a Monday, it will chose the previous 6 to 10 Mondays). Statistically significant changes are then presented, allowing the user to detect temporal changes in traffic behavior.

V. System and Performance

Our system is implemented using a variety of software technologies based on Python, Elixir[6], and Angular[7]. We have also used libraries such as NumPy, Scikit-Learn and

[6] Elixir: https://elixir-lang.org/
[7] AngularJS: https://angularjs.org/



Each Intersection Marker represents the following:
• The size is proportional to the split score observed at the intersection.
• The color is indicative of cluster of the behavioral group the intersection belongs to.
• The shape indicates the potential problems at the intersection, if any

On the left, you can filter Intersections by Split Score and Arrivals on Green to Red ratios (Both Scaled from 0 to 100).

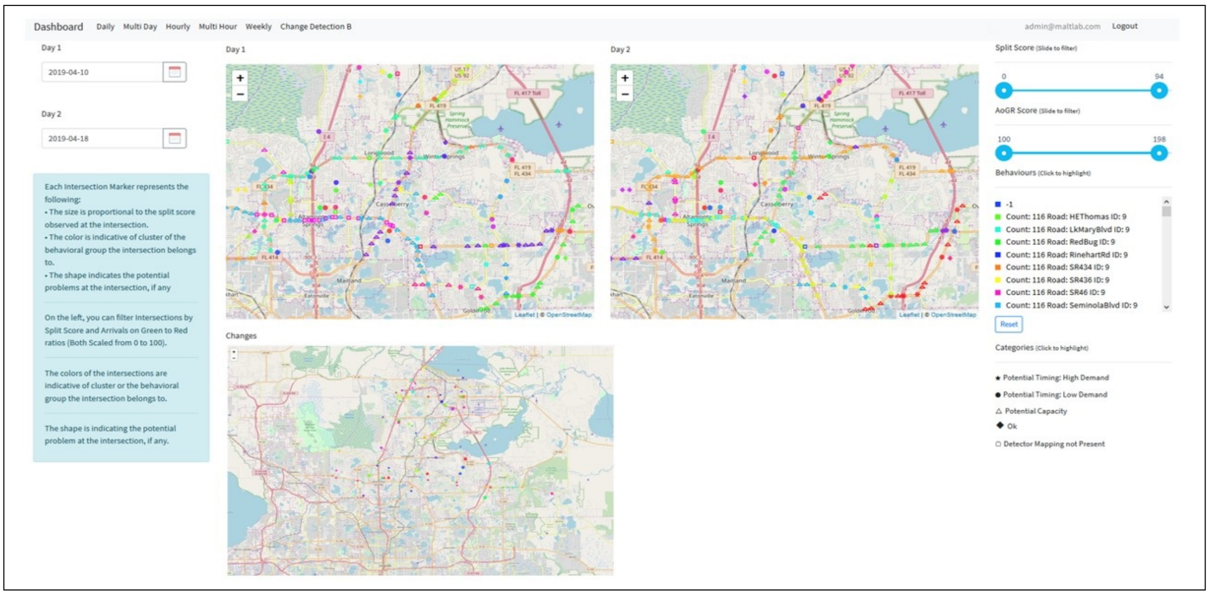
The colors of the intersections are indicative of cluster or the behavioral group the intersection belongs to.

The shape is indicating the

- -1
- Count: 116 Road: HEThomas ID: 9
- Count: 116 Road: LkMaryBlvd ID: 9
- Count: 116 Road: RedBug ID: 9
- Count: 116 Road: SR434 ID: 9
- Count: 116 Road: SR436 ID: 9
- Count: 116 Road: SR46 ID: 9
- Count: 116 Road: SeminolaBlvd ID: 9

★ Potential Timing: High Demand
● Potential Timing: Low Demand
△ Potential Capacity
◆ Ok
▢ Detector Mapping not Present

Fig. 5. Multi-day angular dashboard allows for comparison of the clustering and ranking results across days. It highlights temporal patterns in intersection performance. We can see that the behavior on weekdays can be contrasted with the weekend behavior.


Dashboard
Daily
Multi Day
Hourly
Multi Hour
Weekly
Change Detection B
admin@maltlab.com
Logout
Day 1
Day 2
Changes
Split Score (Slide to filter)
0
94
AoGR Score (Slide to filter)
100
198
Behaviours (Click to highlight)
Reset
Categories (Click to highlight)

Day 1
2019-04-10

Day 2
2019-04-18

Each Intersection Marker represents the following:
• The size is proportional to the split score observed at the intersection.
• The color is indicative of cluster of the behavioral group the intersection belongs to.
• The shape indicates the potential problems at the intersection, if any

On the left, you can filter Intersections by Split Score and Arrivals on Green to Red ratios (Both Scaled from 0 to 100).

The colors of the intersections are indicative of cluster or the behavioral group the intersection belongs to.

The shape is indicating the potential problem at the intersection, if any.

- -1
- Count: 116 Road: HEThomas ID: 9
- Count: 116 Road: LkMaryBlvd ID: 9
- Count: 116 Road: RedBug ID: 9
- Count: 116 Road: RinehartRd ID: 9
- Count: 116 Road: SR434 ID: 9
- Count: 116 Road: SR436 ID: 9
- Count: 116 Road: SR46 ID: 9
- Count: 116 Road: SeminolaBlvd ID: 9

★ Potential Timing: High Demand
● Potential Timing: Low Demand
△ Potential Capacity
◆ Ok
▢ Detector Mapping not Present

Fig. 6. The Change Detection dashboard allows the user to compare and detect statistically significant changes between any two dates or from the baseline behavior.

Pandas. A high level architecture of the system in presented in Figure 7. At the lowest level the monitoring module collects data from a large number of intersections. The module allows for performing collection in real time or at regular intervals. This data is stored in a database. A multi-threaded software layer based on Elixir and Python is used to develop all of our algorithm implementation for each of the modules described in the previous section. This allows for fault tolerance and seamless scalability in presence of additional computational resources. In particular, the architecture has several useful attributes:

- Fault Tolerant: We create a single actor for each functional task, and each actor or thread has it's own supervisor thread. If an actor fails to execute it's task, it suspends itself and all of its children and sends an exception to its supervisor. The supervisor can then work on a recovery strategy. Actors and supervisors fail gracefully, and all the failures can be ultimately managed by the Elixir/Erlang virtual machine (called BEAM).
- Scalable: Because the system has a set of dedicated actors for each functional module and the actors have very little overhead associated with them, it is easy to spawn new actors with an increase in the workload for any specific module. The number of actors is only limited by the resources available on the physical machine.

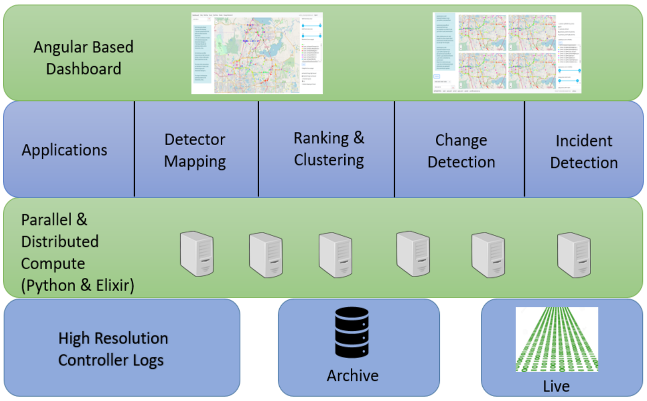


Fig. 7. Overall Architecture of the system. starting with event logs from controllers, we use distributed and parallel compute techniques to design a robust framework to support various applications.

- Parallel and Distributed: This architecture is inherently parallel and can be distributed among multiple machines for further scalability. Any number of nodes running the virtual machine can be merged easily, and the actors or threads running on one node can communicate with threads on other nodes with no extra effort. The only additional overhead is the latency associated with communicating over a network.

The performance of the system was evaluated on a 50-core CPU server. 1 month of controller log data from approximately 1000 intersections was processed in a total of around 180 minutes. The workloads are a function of city size and we have documented batch performance observed on historic data. But, given the performance observed for the current (historic data based) workloads, this system can easily scale for near-real time workloads.

Figures 4, 5 and 6 show a snapshot of a visualization built in Angular 6 (with a Ruby backend) to present our results. The main benefit of choosing Angular is its component-based architecture, which enables the reuse of components and elements across the application. Also, the use of services in Angular assists in sharing the data across components with similar functionality.

The maps that are embedded in the application are built using leaflet.js. Leaflet is a JavaScript library that provides interactive maps and contains all the required mapping features. Since the application deals with a large number of intersections, it is important for both scalability and responsiveness to show these intersections as markers on the map without any noticeable lag. For example, whenever the user selects a range of dates, more than 10,000 markers are plotted on the maps. Making the markers are made with SVG or HTML reduces the performance of the application because all the markers have to be loaded into the DOM (Document Object Model). To overcome this issue, each marker is drawn using Canvas, and because Canvas markers need not be loaded into the DOM the maps can handle more than 100,000 markers at once without sacrificing user interactivity.

VI. Conclusions

The advent of new traffic data logging, collection, and reporting tools enhances traffic signal management by using these high resolution controller logs to generate newer operational performance measures. However, it can be challenging to continuously monitor these performance measures for even small cities (comprising a few hundred intersections). In this work, we presented a scalable system that provides corridor-level and city-level information using these measures in a succinct and actionable form. Specifically, we ranked and categorized intersections by using the two measures, allowing users to filter intersections based on these measures and categories, enabling them to focus on key problem areas. We grouped intersections into clusters of similar performance to enable detection of spatial and temporal patterns in performance and to aid the detection of changes in performance. Our system leverages high performance computing and machine learning methodologies for data collected from a large number of intersections to derive key spatio-temporal traffic patterns in a city. It allows a traffic engineer to interactively understand problems and focus on key challenges or improvements that need to be carried out to alleviate traffic congestion in a city.

Acknowledgments

This work was supported by NSF CNS 1922782 and by the Florida Dept. of Transportation (FDOT). The opinions, findings and conclusions expressed in this publication are those of the author(s) and not necessarily those of the Florida Department of Transportation or the National Science Foundation.

References

[1] C. Day, D. Bullock, H. Li, S. Remias, A. Hainen, R. Freije, A. Stevens, J. Sturdevant, and T. Brennan, "Performance measures for traffic signal systems: An outcome-oriented approach," *Joint Transportation Research Program*, 01 2014.
[2] US Department of Transportation, Federal Highway Administration, "Traffic signal timing manual," https://ops.fhwa.dot.gov/publications/fhwahop08024/index.htm, 06 2008, (Accessed on 2/10/2020).
[3] ——, "Measures of effectiveness and validation guidance for adaptive signal control technologies," https://ops.fhwa.dot.gov/publications/fhwahop13031/index.htm, 07 2013, (Accessed on 2/10/2020).
[4] D. Mahajan, Y. Karnat, T. Banarjee, A. Rangarajan, and S. Ranka, "A data driven approach to derive traffic intersection geography using high resolution controller logs," 2020, in press, 6th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS.
[5] D. Mahajan, T. Banerjee, A. Rangarajan, N. Agarwal, J. Dilmore, E. Posadas, and S. Ranka, "Analyzing traffic signal performance measures to automatically classify signalized intersections," in *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS*, INSTICC. SciTePress, 2019, pp. 138–147.
[6] D. Mahajan, Y. Karnati, A. Rangarajan, and S. Ranka, "Unsupervised summarization and change detection in high-resolution signalized intersection datasets," submitted to 2020 IEEE Intelligent Transportation Systems Conference (ITSC).
[7] Y. Karnati, D. Mahajan, A. Rangarajan, and S. Ranka, "Data mining algorithms for traffic interruption detection," 2020, in press, 6th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS.