TITAN – AN INTERACTIVE WEB-BASED PLATFORM FOR TRANSPORTATION DATA INTEGRATION AND ANALYTICS

DATA CENTER
Query Crash Data
Search and query data from our databases
QUERY
Detector Data
Get Statewide Detector Data
QUERY
Probe Data
Get Probe Data
QUERY
Upload Data
Upload Data - Get Interactive Apps
SHARE

May 2019
Final Report

Project number TR201815
MoDOT Research Report number cmr 19-006

PREPARED BY:

Carlos Sun, Ph.D., P.E., J.D (P.I.)

Praveen Edara, Ph.D., P.E. (co-P.I.)

Yaw Adu-Gyamfi, Ph.D. (co-P.I.)

University of Missouri-Columbia Department of Civil and Environmental Engineering

PREPARED FOR:

Missouri Department of Transportation

Construction and Materials Division Research Section

TITAN – An Interactive Web-based Platform for Transportation Data InTegration and ANalytics

Principal Investigator
Carlos Sun, Ph.D., P.E., J.D
Professor
Department of Civil and Environmental Engineering
University of Missouri-Columbia

Co-Principal Investigator
Praveen Edara, Ph.D., P.E.
Professor
Department of Civil and Environmental Engineering
University of Missouri-Columbia

Co-Principal Investigator
Yaw Adu-Gyamfi, Ph.D.
Assistant Professor
Department of Civil and Environmental Engineering
University of Missouri-Columbia

A Report on Research Sponsored by

Missouri Department of Transportation

May 2019

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. cmr 19-006	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle TITAN – An Interactive Web-based Platform for Transportation Data InTegration and ANalytics		**5. Report Date** May 2019 Published:May 2019 **6. Performing Organization Code** 00062450
7. Author(s) Carlos Sun, Ph.D., P.E., J.D. https://orcid.org/0000-0002-8857-9648 Yaw Adu-Gyamfi, Ph.D. https://orcid.org/0000-0002-1924-9792 Praveen Edara, Ph.D., P.E. https://orcid.org/0000-0003-2707-642X		**8. Performing Organization Report No.**
9. Performing Organization Name and Address University of Missouri Department of Civil & Environmental Engineering E2509 Lafferre Hall Columbia, Missouri 65211		**10. Work Unit No.** **11. Contract or Grant No.** MoDOT project #TR201815
12. Sponsoring Agency Name and Address Missouri Department of Transportation (SPR-B) Construction and Materials Division P.O. Box 270 Jefferson City, MO 65102		**13. Type of Report and Period Covered** Final Report (June 2018-May 2019) **14. Sponsoring Agency Code**
15. Supplementary Notes Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration. MoDOT research reports are available in the Innovation Library at https://www.modot.org/research-publications.		
16. Abstract The Missouri Department of Transportation regularly collects and stores various types of data for different uses such as planning, traffic operations, design, and construction. These large datasets contain treasure troves of information that could be fused and mined, but the size and complexity of data mining requires the use of advanced tools such as Big Data Analytics, machine learning, and cluster computing. TITAN is an initial prototype of an interactive web-based platform developed to demonstrate the possibilities of using big data software. This project succeeded in showing a user-friendly front end that was graphical in nature and a scalable back end that was capable of integrating multiple big databases with no latency. Several applications were shown, including mobility, safety, integrated safety and mobility, detectors, and predictive crash analytics. Because TITAN was shown to be feasible and efficient, the next step is to deploy TITAN to assist MoDOT staff in various data-driven decision-making processes.		
17. Key Words Data analysis; Data fusion; Databases; Prototypes; Web applications. Big Data Analytics, Visualization, Cluster computing	**18. Distribution Statement** No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.	

19. Security Classif. (of this report) Unclassified.	**20. Security Classif. (of this page)** Unclassified.	**21. No. of Pages** 60	**22. Price**

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Table of Contents

List of Figures

List of Tables

Acknowledgments

The researchers acknowledge the support provided by our research partners who provided guidance and ideas. Some of these partners include Cheryl Ball, MoDOT Waterways & Freight Administrator, Randy Johnson, MoDOT KC Scout Traffic Center Manager, Ray Webb, Mid-America Regional Council Manager of Traffic Operations, and Brian Umfleet, MoDOT Gateway Guide Traffic Center Manager.

Copyright

Disclaimer

Executive Summary

An exponential growth in relevant data streams (e.g. traffic, geometric, safety, pavements, crowd-sourced) has brought new opportunities and challenges in the realm of transportation data warehousing. Increased data enables improved planning, monitoring, prediction, and management of transportation systems, but only if the manipulation of such gigantic datasets could be automated efficiently. With the increasing demand for modern data warehousing, there has been a significant growth in commercial and open-source tools. TITAN was developed utilizing a completely open-source platform that leverages recent advances in big data analytics to efficiently process multiple streams of data and deploy applications that will enable transportation personnel to make practical, data-driven decisions. TITAN has two main components: a data center and an applications center. The data center stores different streams of datasets and provides a user-friendly interface for querying the different databases. By leveraging cluster computing and recent advances in big data analytics, TITAN is able to generate responses to different forms of user queries at a much faster rate compared to traditional data warehouses.

The second component of TITAN is the APPCENTER (Applications Center) which hosts a variety of applications for performance monitoring, data integration and predictive analytics. The APPCENTER is powered with fast, interactive visualizations that enable users to identify trends and insights quickly for decision making. The APPCENTER was developed on top a Graphical Processing Unit (GPU) database which enables it to perform computations on large datasets within fractions of a second. Examples of applications in the APPCENTER include crash risk prediction, safety-mobility performance measures, and traffic surveillance, as shown in Figure ES.1.

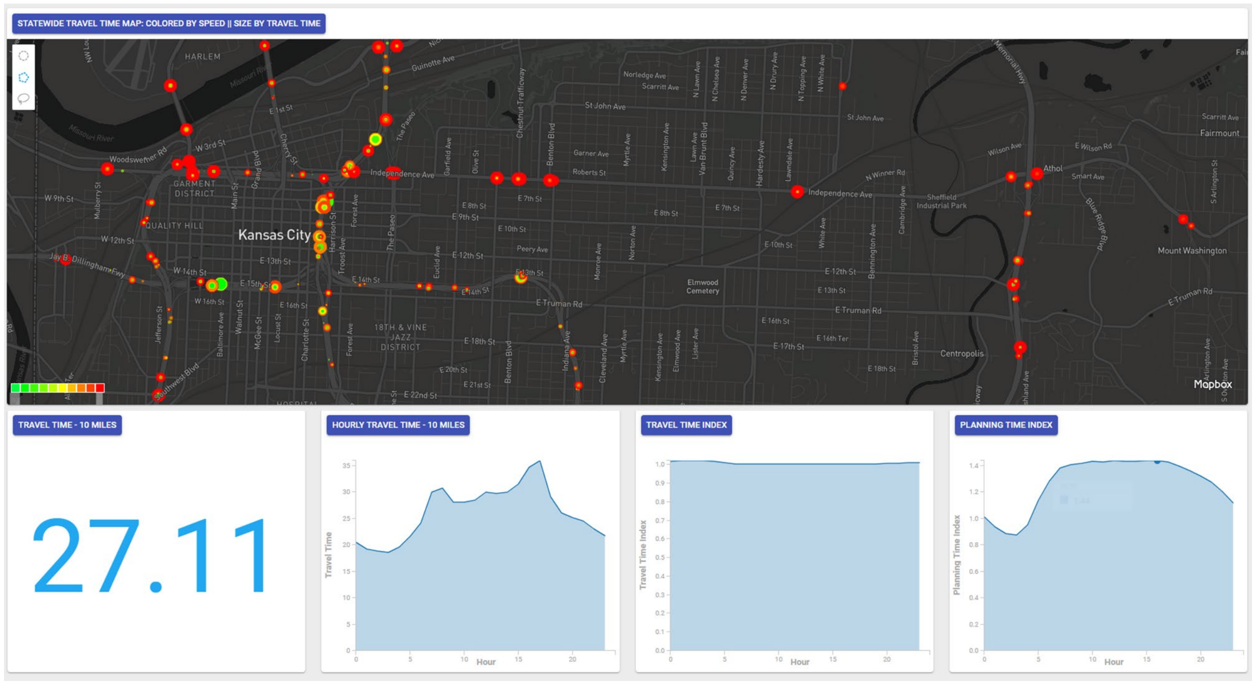


Figure ES.1. Web-Based Interactive Visualization and Analytics

Figure ES.2 summarizes the unique system and analytical capabilities of the TITAN platform. The customized dashboards developed in TITAN can be used to communicate performance measures on different transportation infrastructure or assets. TITAN can serve both real time and static dashboards, depending on agency needs. As data streams are ingested into TITAN, the dashboards will be automatically updated. The platform is enabled with graphical processing units (GPUs) and machine learning algorithms to power predictive analytics.

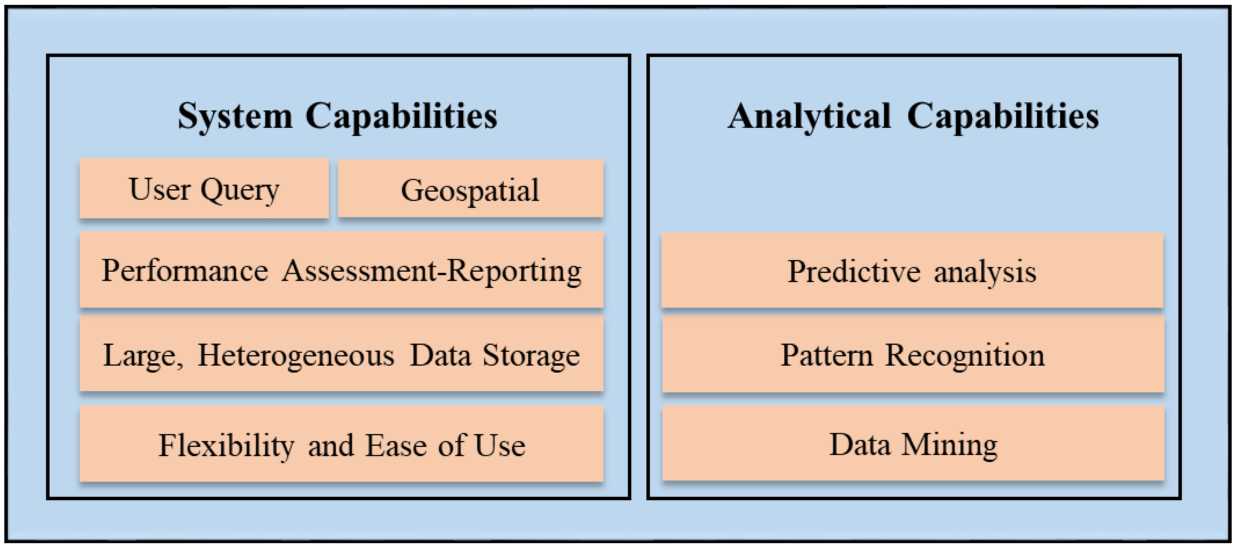


Figure ES.2. Analytical and System Capabilities of TITAN

A public agency, like the Missouri Department of Transportation (MoDOT), compiles and stores large amounts of data from various sources, both public and commercial. For example, automated road analyzer (ARAN) vans collect pavement condition information, count stations and counters gather traffic volumes, the Missouri Highway Patrol compiles crash reports, and third-party vendors assemble crowd-sourced GPS-located data. Traditionally, each source of information served a limited number of purposes, sometimes one only. For example, location data has been used to estimate travel times and traffic volumes to estimate annual average daily traffic. But what if all these types of information were tightly integrated together so that a particular application, project or location can utilize all relevant databases? An application like TITAN allows MoDOT to harness the wealth of existing data sources and to do so efficiently and easily via graphical tools. As TITAN is a prototype software, this final report also serves as a user manual.

Chapter 1 Introduction

1.1. Background

The recent surge in the use of community-based sensing such as Waze, ubiquitous mobile computing, automatic vehicle location equipped fleets, surveillance cameras, and so on has exponentially increased the rate of data collection for most transportation agencies. Increased monitoring of transportation networks, however, will only be fruitful if timely analysis can be conducted to provide actionable insights needed for the day to day management of the transportation system.

Although agencies such as State Departments of Transportation (DOTs) have transportation data management systems for storing and processing data streams, they are not uniquely designed to handle such large, heterogeneous, and multi-resolution data streams. They have limited analytical capabilities that will enable them to integrate, mine, visualize and predict on large, multivariate datasets at reasonable speeds (Mostafa et al. 2018, Richardson et al. 2014). Traditional data warehouses are stretched to the limit due to the enormous size and speed, and the significant variety of datasets across different vendors in terms of collection method, data quality, availability (daily, monthly or quarterly), and format (shapefiles, documents, table, videos, etc.). The need for frameworks or platforms that can quickly integrate, digest and extract actionable insights from these datasets is therefore crucial.

Recent advances in Big Data Analytics are enabling organizations to digest humongous amounts of data and transform them into actionable insights (Kluger et al, 2013, Adu-Gyamfi et al, 2016). This innovation is being fueled by massive open data platforms, driven by machine learning and empowered by low cost cloud computing. This new wave of invention could be leveraged to enable transportation agencies to identify the usefulness of their diverse datasets and to explore previously untapped applications.

1.2. Project Objectives

Under the above context, the primary goal of this project is to deliver a prototype design and deployment of TITAN, an interactive, web-based platform that will assist decision makers at MoDOT to seamlessly integrate and analyze its transportation datasets. The prototype platform is designed to be significantly faster and cheaper (by using open-sourced software solutions for development) compared to conventional data warehouses, which are heavily reliant on relational databases housed in big, costly enterprise machines. The four key sub objectives that arise from the primary goal are:

1. Leverage state-of-the-art big data frameworks to develop a platform that is fast and scalable for data analytics unlike traditional data warehouses commonly used by transportation agencies.
2. Offer a low-cost but effective data integration and analytics platform by leveraging open-source software for designing, developing and deploying TITAN.
3. Provide user-centered, web-based data visualization to allow for easy interaction with the platform.
4. Provide users quick access to the integrated datasets via a user-friendly, interactive web interface.

1.3. Report Organization

The remainder of this report is organized as follows: Chapter 2 covers the design of the key components of TITAN. It covers the frameworks used: the software and databases deployed at a high level. In Chapter 3, we explain how users can upload and download data from TITAN via a web interface. Chapter 4 highlights the Applications Center (APPCENTER), which is the heart of the TITAN platform. A variety of apps for performance monitoring and predictive analytics are covered in this section. A pipeline on how MoDOT could integrate the key outcomes, recommendations and limitations of the project is presented.

Chapter 2 Titan Development

Figure 2.1 illustrates the design framework that enables TITAN to handle fast, disparate, noisy data streams in a seamless manner. It comprises of two key modules: first, a user-centered, interactive, web-based front end where TITAN's users can interface and interact with the platform and second, a big data enabled back end which stores data and provides a computational framework for retrieving, processing and visualizing large datasets. TITAN's design architecture seeks to address key technological gaps—in data handling, archiving and analysis for decision support. The following section provides a detailed overview of the components of TITAN.

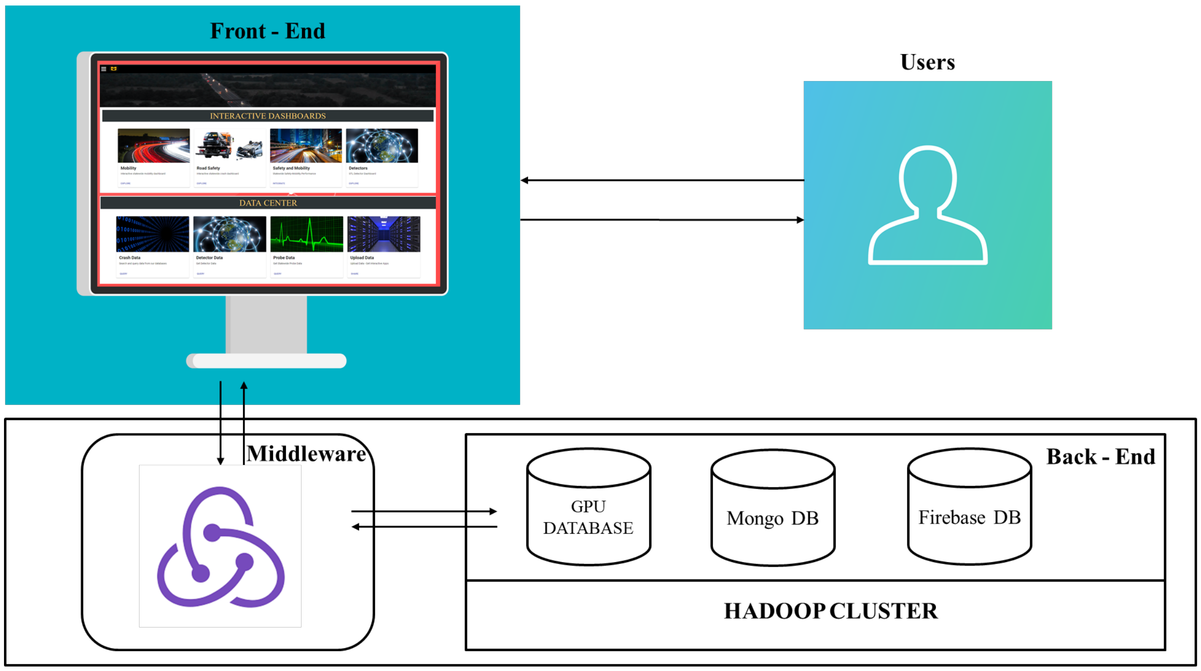


Figure 2.1. Schematic of TITAN's Key Components

2.1. Front-End Module Design

In order to be user-friendly, the front end of TITAN helps to mask software specifications and requirements by using a variety of layouts and user interface (UI) elements. This enables users to interact with applications directly on any computer, with any browser, and from any location with internet access. The main challenge associated with front end module designs is that the tools and frameworks used to create them change constantly. As a result, a key consideration for selecting a front end development framework includes both the number of developers contributing to the development of libraries and its popularity among developers. Table 2.1 shows a list of the most popular front end development frameworks, the number of contributors and its popularity among developers using the GitHub popularity metric. GitHub is a web-based computer code hosting service that is especially popular with open-source software projects.

Table 2.1. Choosing a Front End Library for TITAN's Development

Front End Library	Number of Contributors	GitHub Popularity
React	1285	124,747
Aurelia	97	10,873
Angular	1596	59,434
Ember	753	20,772
Vue	268	131,290

TITAN was developed with React (Todd, 2016), a JavaScript library developed by Facebook for building interactive user interfaces. React is the second most popular front-end development framework with about 125,000 stars on GitHub. With 1285 active contributors and increasing, the library is able to catch up with the constantly-changing requirements for front end development. Although Vue is the most popular framework, it was not used because it is relatively new and has fewer developers contributing to its libraries.

There are 2 main components of TITAN that users can access via the front end user interface: the Applications Center (APPCENTER) and the Data Center. The following section provides a summary of these components, starting with the need to register an account.

2.1.1 Registration and Login

To take advantage of the full capabilities of TITAN, all first time users of the platform must register with an email and password via a log-in or registration screen. The user will only be required to go through the log-in process once if the same computer is used to access TITAN. The credentials of each user are stored in a database to manage user activities. Figure 2.2 shows the login or register screen.

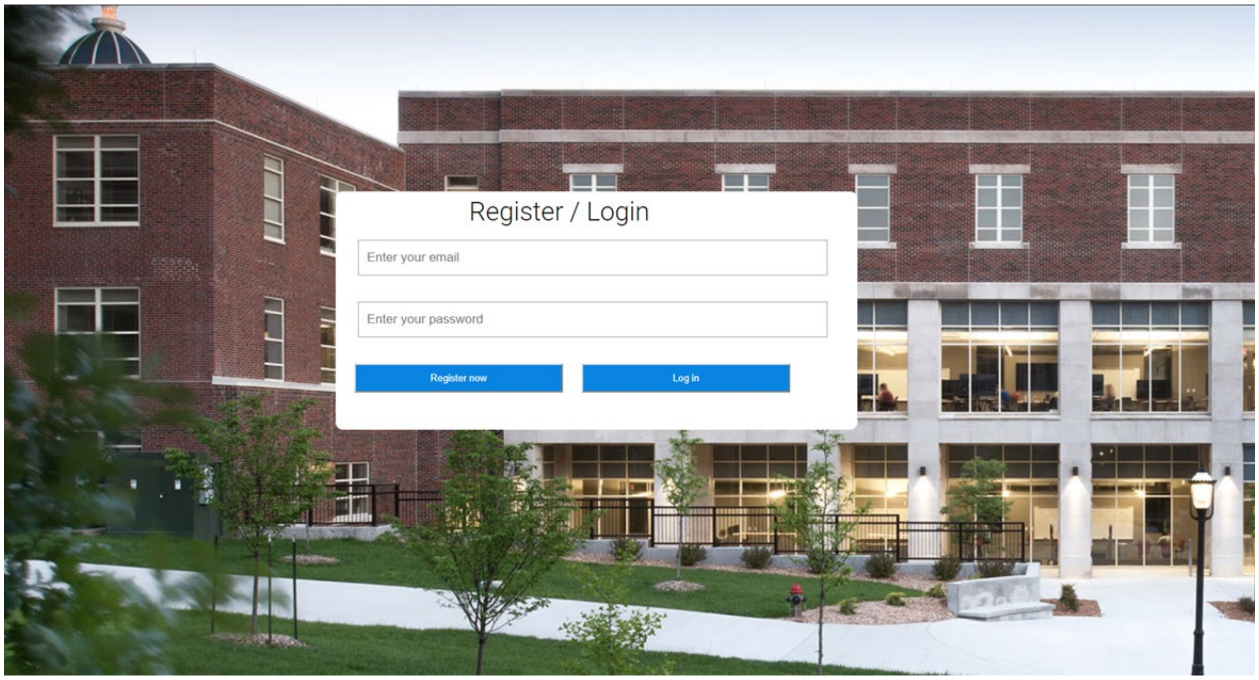


Figure 2.2. User Registration and Login

The authentication framework, shown in figure 2.3, was developed to manage the types of modules that a user could access based on their credentials. In its current state, the APPCENTER is only accessible to users who are registered. Other modules such as data querying and visualization are accessible to the general public. Depending on the requirements of the transportation agency, the logic for user authentication can be made more or less secure.

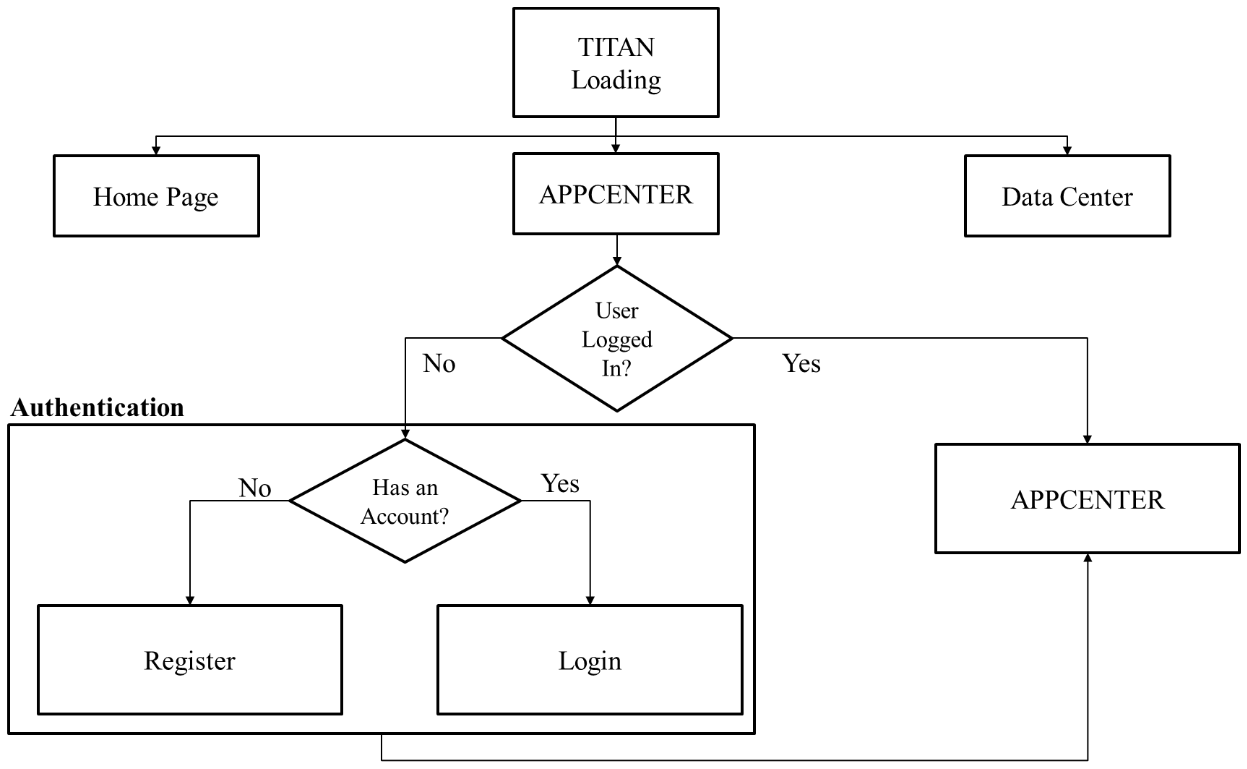


Figure 2.3. TITAN Authentication Framework.

2.2 Back-End Module Design

The primary goal of TITAN's Back-End is to provide computational resources that could be used to speed up responses to user queries from the front end. The types of analytics carried out on TITAN's front end can be computationally expensive. For example, a user may request to calculate the average travel time during PM peak hours on all major arterials in St. Louis over a period of 5 years. It is expected that the back end is able to sift through approximately 100 gigabytes of data and provide a response to the user without any significant latencies. To enable TITAN to carry out such highly complex analytics from the front end, we designed a scalable, cloud-based back end based on recent advances in big data analytics. In addition to providing computational resources, the back end is also used for archiving large datasets and managing functions such as user authentication, data security, etc. The structure of the back end and how it integrates into the overall TITAN framework is shown in figure 2.4. At the core of the back end is a Hadoop distributed file system (HDFS) (Shvachko et al. 2010) which enables the networking of a series of computers into cluster. By using the HDFS, we are able to maintain the processing speed of TITAN even as the size of data grows. On top of the Hadoop framework are three different databases: Firebase (Singh et al. 2018), MongoDB (Chodorow, 2013) and a GPU database. The roles of each of these databases are defined as follows.

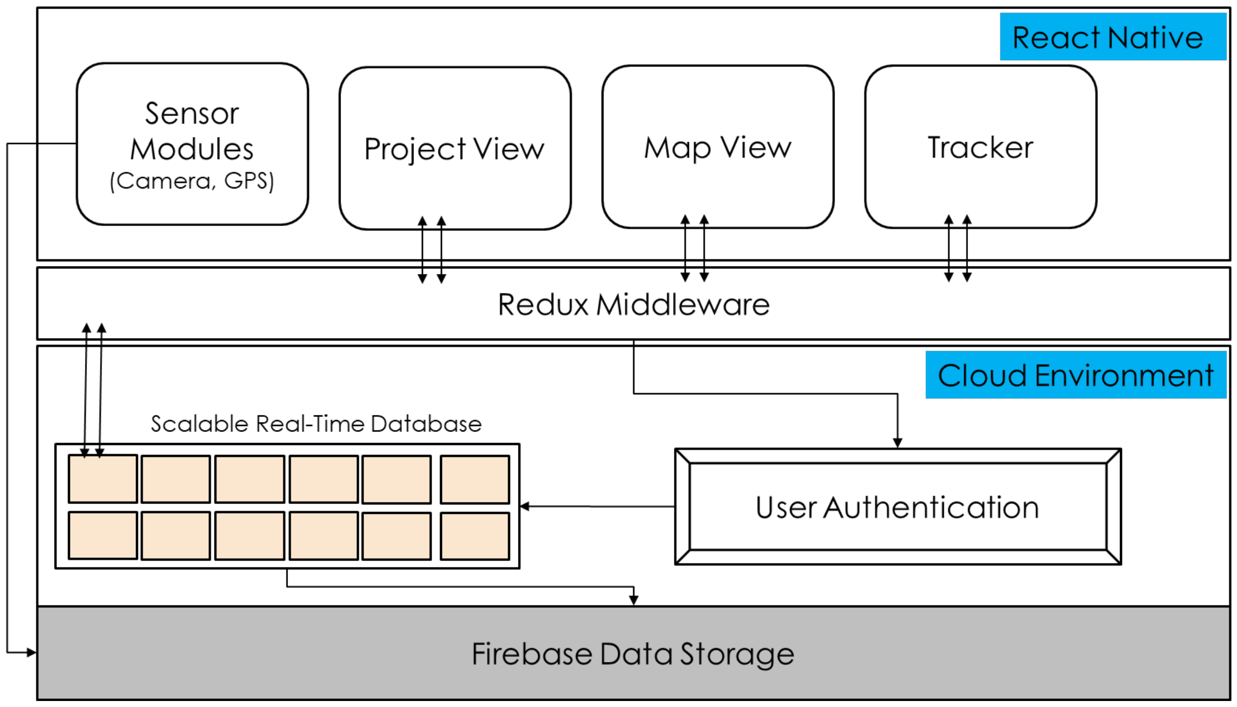


Figure 2.4. Back End Integration into TITAN's Framework.

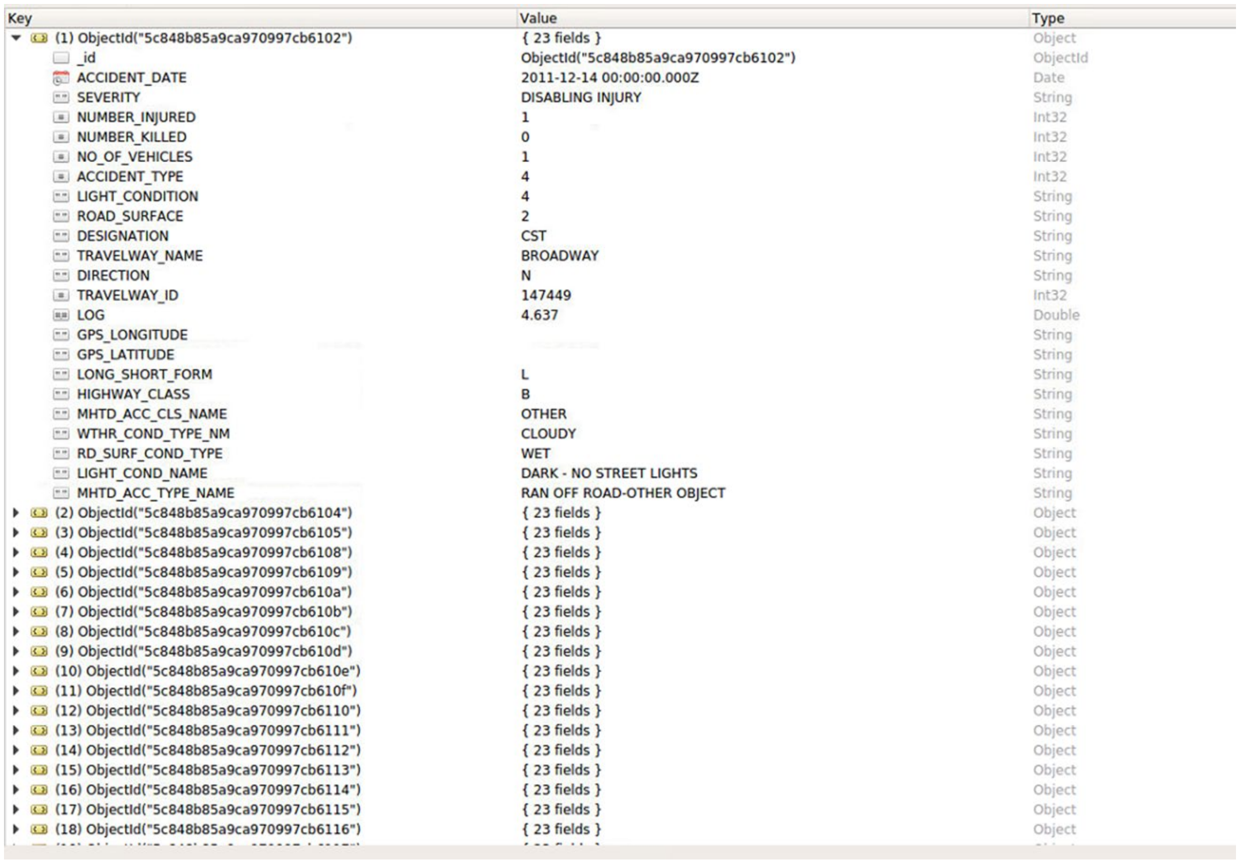
2.2.1 Firebase DB

Firebase is a NOSQL database for storing datasets that are not structured. NOSQL means non-relational in contrast to traditional SQL databases. NOSQL also differs in how it is scaled by increasing the database server pool as opposed to increasing the horsepower of the hardware.

Firebase serves two functions in the TITAN framework: user authentication and temporary data storage. Before a user can use the app, they must be authenticated. Firebase authenticates users via email/password, phone number, or Facebook, Google, Twitter, and Github accounts. To simplify TITAN's development, users are authenticated only by email and password. Firebase is also used to temporarily store all user uploaded datasets. Since Firebase is a NOSQL database, it is able to consume all types of data formats: Shapefiles, CSVs, XML, Video, Images, etc. In contrast, a SQL database requires a predefined format or schema. It is however, not designed for storing very large datasets. Firebase therefore redirects significantly large files (30GB or more) to the HDFS and MongoDB depending on the use of the data.

2.2.2 Mongo DB

Mongo DB is also a NOSQL database. Compared to Firebase, Mongo is highly scalable and as such can handle much larger files. TITAN uses Mongo primarily for managing data queries at the Data Center. Figure 2.5 shows an example of how data is stored in Mongo. Each row of data is stored uniquely with an identifier (_id). Because data is stored by rows and not as one big table, Mongo can store files with uneven number of columns. This is a relevant property for handling unstructured datasets such as texts, images, videos, etc.



Key	Value	Type
(1) ObjectId("5c848b85a9ca970997cb6102")	{ 23 fields }	Object
_id	ObjectId("5c848b85a9ca970997cb6102")	ObjectId
ACCIDENT_DATE	2011-12-14 00:00:00.000Z	Date
SEVERITY	DISABLING INJURY	String
NUMBER_INJURED	1	Int32
NUMBER_KILLED	0	Int32
NO_OF_VEHICLES	1	Int32
ACCIDENT_TYPE	4	Int32
LIGHT_CONDITION	4	String
ROAD_SURFACE	2	String
DESIGNATION	CST	String
TRAVELWAY_NAME	BROADWAY	String
DIRECTION	N	String
TRAVELWAY_ID	147449	Int32
LOG	4.637	Double
GPS_LONGITUDE		String
GPS_LATITUDE		String
LONG_SHORT_FORM	L	String
HIGHWAY_CLASS	B	String
MHTD_ACC_CLS_NAME	OTHER	String
WTHR_COND_TYPE_NM	CLOUDY	String
RD_SURF_COND_TYPE	WET	String
LIGHT_COND_NAME	DARK - NO STREET LIGHTS	String
MHTD_ACC_TYPE_NAME	RAN OFF ROAD-OTHER OBJECT	String
(2) ObjectId("5c848b85a9ca970997cb6104")	{ 23 fields }	Object
(3) ObjectId("5c848b85a9ca970997cb6105")	{ 23 fields }	Object
(4) ObjectId("5c848b85a9ca970997cb6108")	{ 23 fields }	Object
(5) ObjectId("5c848b85a9ca970997cb6109")	{ 23 fields }	Object
(6) ObjectId("5c848b85a9ca970997cb610a")	{ 23 fields }	Object
(7) ObjectId("5c848b85a9ca970997cb610b")	{ 23 fields }	Object
(8) ObjectId("5c848b85a9ca970997cb610c")	{ 23 fields }	Object
(9) ObjectId("5c848b85a9ca970997cb610d")	{ 23 fields }	Object
(10) ObjectId("5c848b85a9ca970997cb610e")	{ 23 fields }	Object
(11) ObjectId("5c848b85a9ca970997cb610f")	{ 23 fields }	Object
(12) ObjectId("5c848b85a9ca970997cb6110")	{ 23 fields }	Object
(13) ObjectId("5c848b85a9ca970997cb6111")	{ 23 fields }	Object
(14) ObjectId("5c848b85a9ca970997cb6112")	{ 23 fields }	Object
(15) ObjectId("5c848b85a9ca970997cb6113")	{ 23 fields }	Object
(16) ObjectId("5c848b85a9ca970997cb6114")	{ 23 fields }	Object
(17) ObjectId("5c848b85a9ca970997cb6115")	{ 23 fields }	Object
(18) ObjectId("5c848b85a9ca970997cb6116")	{ 23 fields }	Object

Figure 2.5. Data Storage in Mongo DB

2.2.3 Graphical Processing Unit (GPU) DB

All front-end visualizations are carried out by using an open-source Graphical Processing Unit (GPU) database. GPUs have tremendous processing capabilities compared to CPUs. For example, a single NVIDIA Geoforce 1080ti GPU card has over 4000 processing cores, meaning 1 GPU is capable of processing information at a rate comparable to using a cluster of 500 8-core computers. The GPU database is the reason why TITAN is so fast on the front end. A Structured Query Language (SQL) database is used on top of this database to process data in the memory of the Graphical Processing Unit (GPU).

Chapter 3 Data Center

In the Data Center, TITAN provides users a user-friendly non-programmatic interface for querying and uploading data from multiple sources at a fast rate. The design components of the Data Center are shown in figure 3.1. The main resources used include a Hadoop cluster which stores all the different datasets in different formats. MongoDB is used to restructure and simplify each data type so that it can be made available in formats such as CSVs and XMLs. Firebase is used as an initial storage engine when files are uploaded before they are pushed to the cluster.

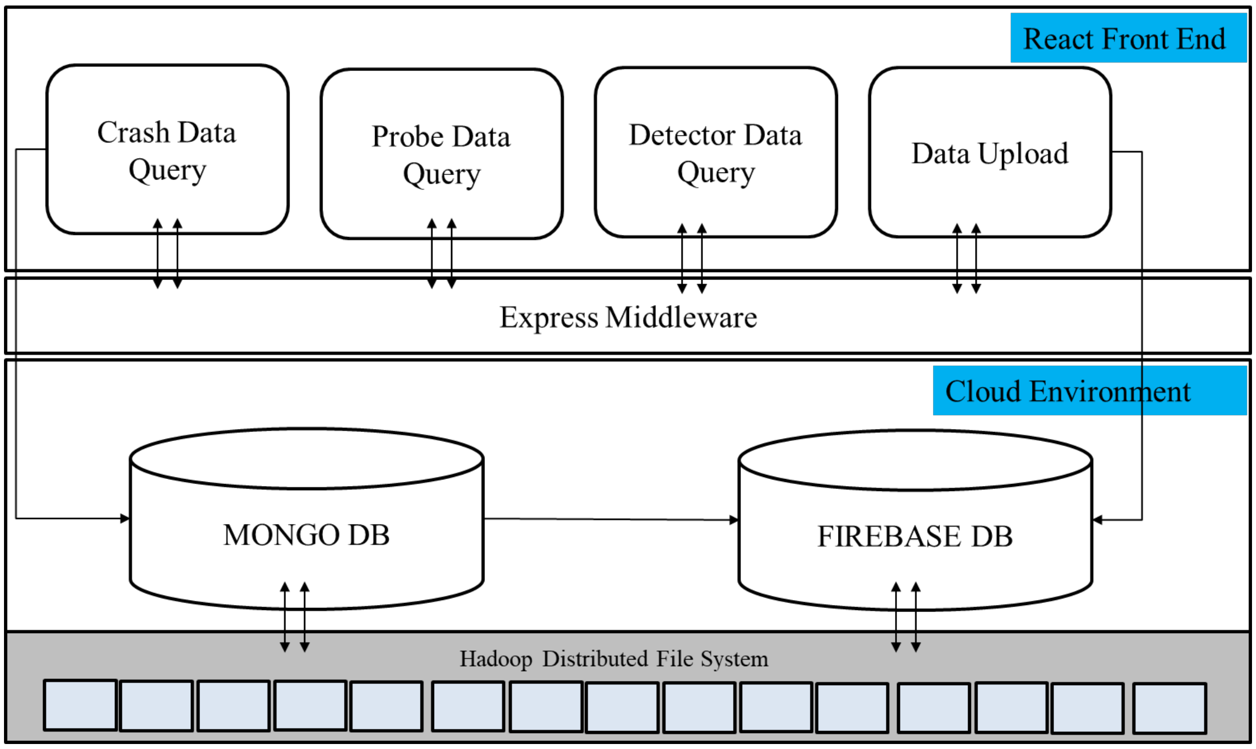


Figure 3.1. Design Components of Data Center

In its current state, the TITAN Data Center provides access to three different databases: a statewide crash, probe and traffic detector databases. It also hosts a single application for uploading different types of data into TITAN's databases. Figure 3.2 shows all the applications deployed in the Data Center.

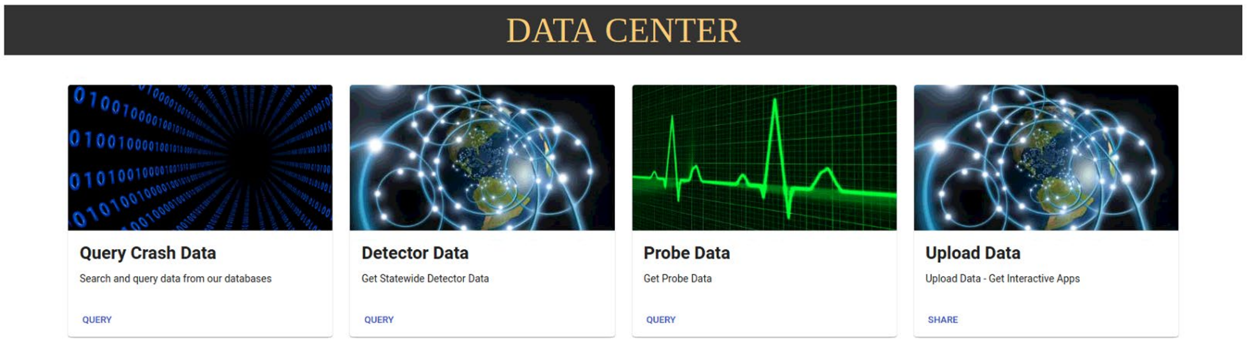

DATA CENTER
Query Crash Data
Search and query data from our databases
QUERY
Detector Data
Get Statewide Detector Data
QUERY
Probe Data
Get Probe Data
QUERY
Upload Data
Upload Data - Get Interactive Apps
SHARE

Figure 3.2. Data Center

3.1 Data Upload

Sharing data on TITAN is a straightforward process. A layout of the form for uploading data into TITAN is shown in figure 3.3. The information requested helps to build applications that are driven by user or agency input. Users are required to provide the name of the agency sharing the data, the type of data, possible uses and limitations of the data. Once data is submitted, it is manually reviewed and a ticket is sent for application development. The interface accepts all types of data including Shapefiles, CSVs, Excel, API links, etc.

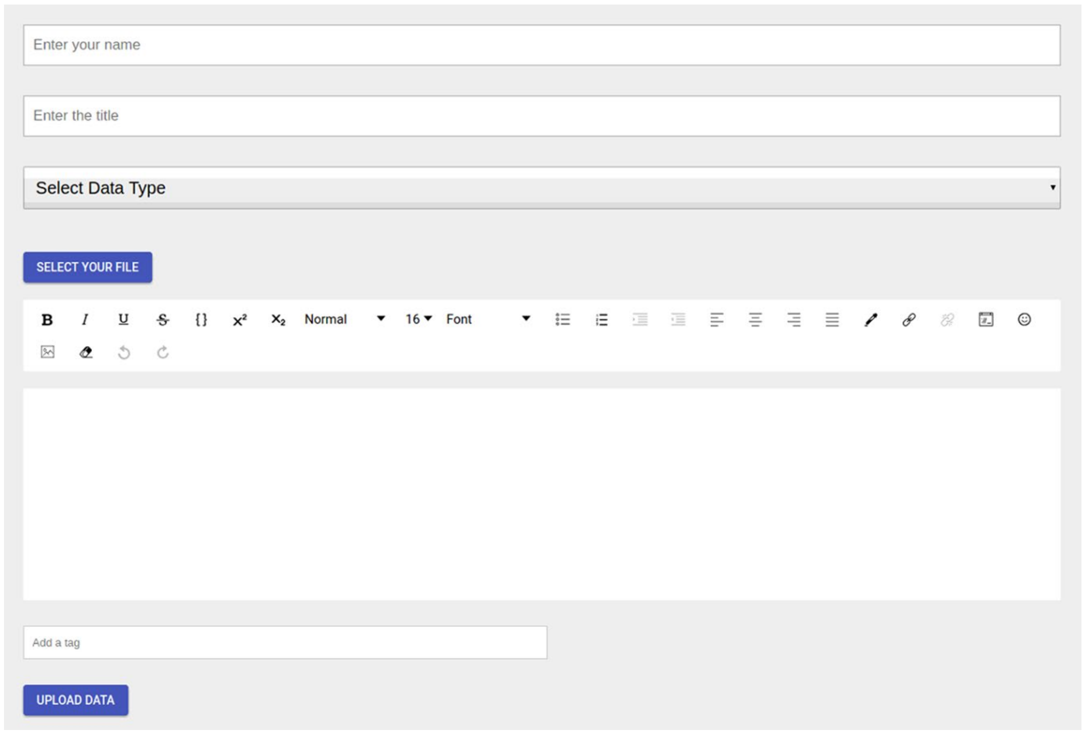
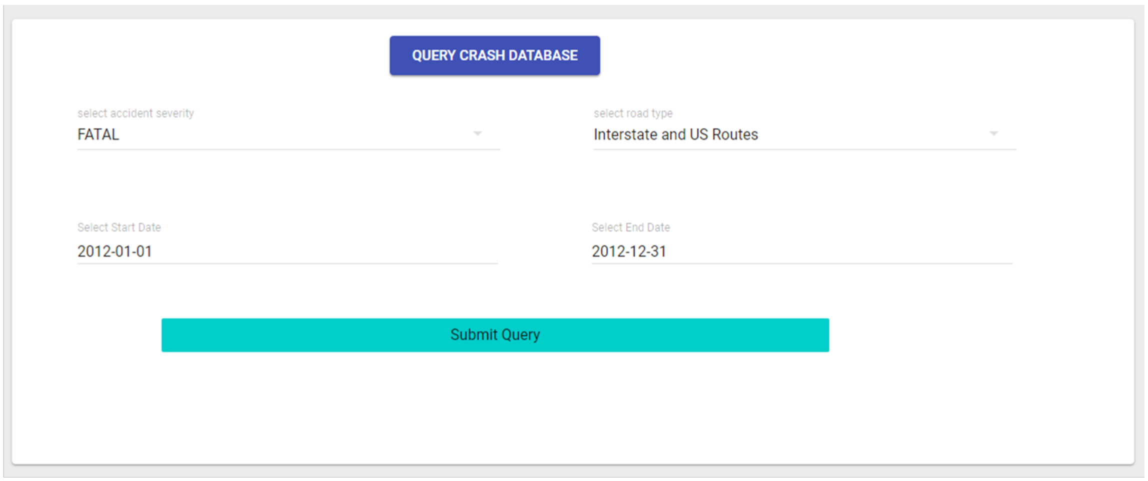


Figure 3.3. Uploading Data to TITAN

3.2 Data Query

The Data Center also permits users to query data from different sources such as crash, detector and probe data. The non-programmatic interface is enabled with functions that help users to select and filter their respective areas of interest. The output of each query is a downloadable comma-delimited (CSV) file which contains all the information requested by the user. The simplicity of the CSV file allows data to be inputted into wide range of software such as spreadsheets and database clients. Figure 3.4 shows an example interface for querying crash data. Due to the scalable design architecture used to develop the Data Center, there are no restrictions on the amount of data that can be queried from the databases. The time needed to respond to a submitted query, however, depends on the amount of data requested. Figure 3.5 shows query response times for different types of crash data requests. The main factors marginally affecting query response times are the aggregation interval and the total length of data requested. Similar charts for detector and probe data query interfaces are shown in figures 3.6 through 3.9.



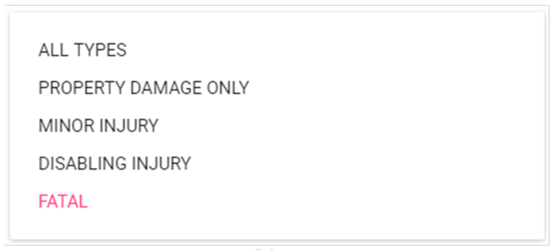
select accident severity
FATAL

select road type
Interstate and US Routes

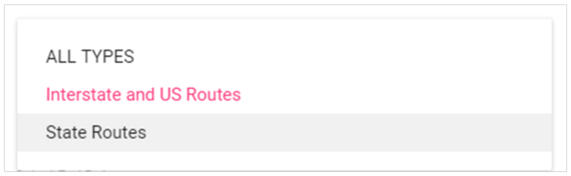
Select Start Date
2012-01-01

Select End Date
2012-12-31

a).

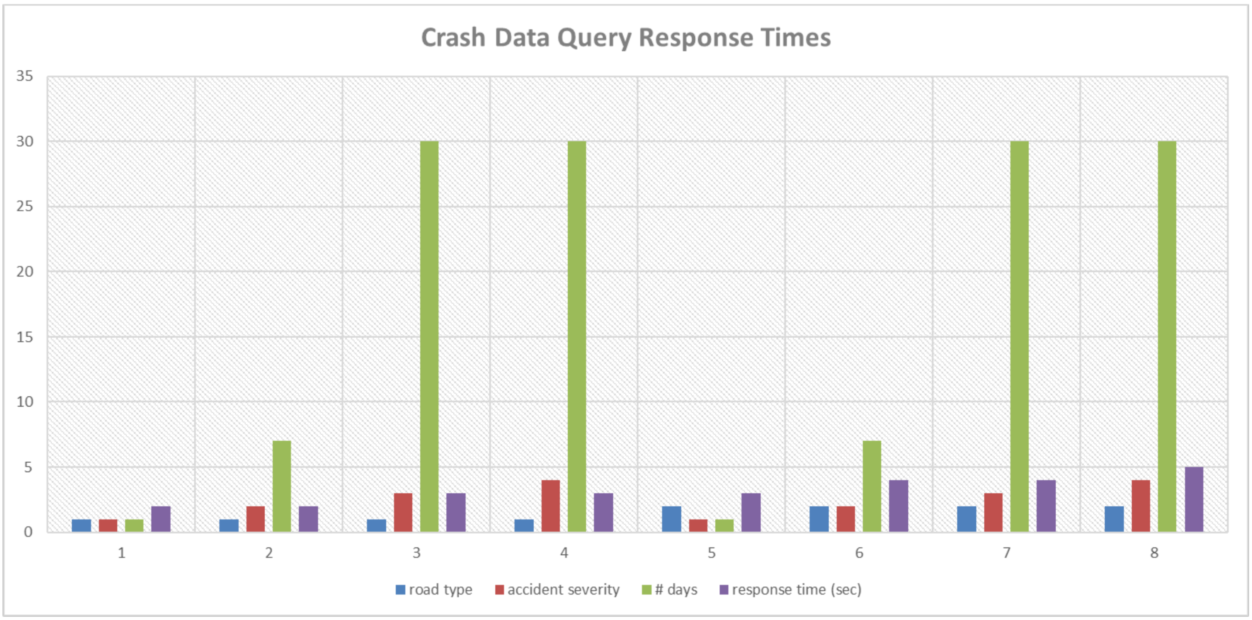


b).



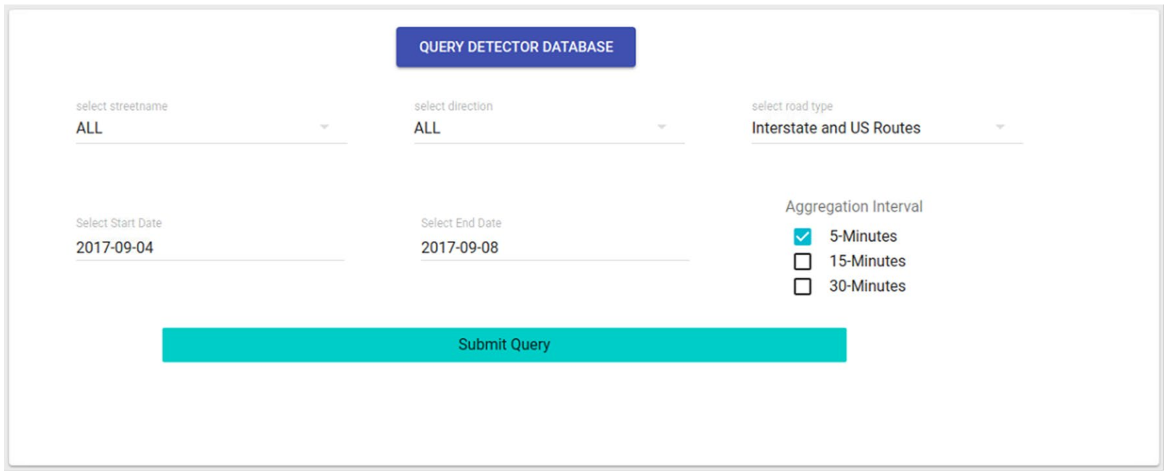
c).

Figure 3.4. Crash Data Query Interface

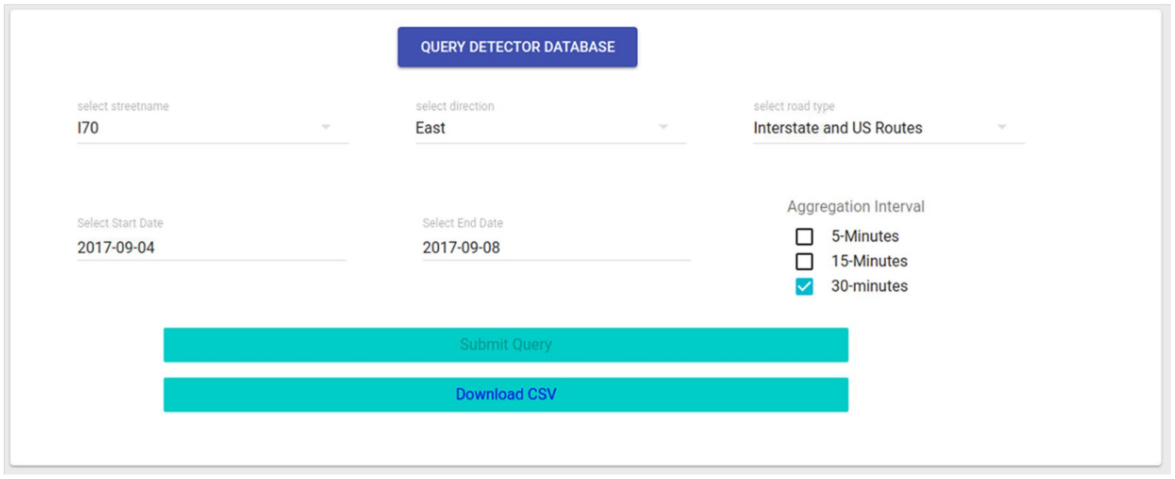


	1	2	3	4	5	6	7	8
road type	1	1	1	1	2	2	2	2
accident severity	1	2	3	4	1	2	3	4
# days	1	7	30	30	1	7	30	30
response time (sec)	2	2	3	3	3	4	4	5

Figure 3.5. Crash Database Query Times: Road Types – Freeways (1), Interstates (2), All Road Types (3). Accident Severity – Fatal (1), Disabling Injury (2), Minor Injury (3), Property Damage (4).

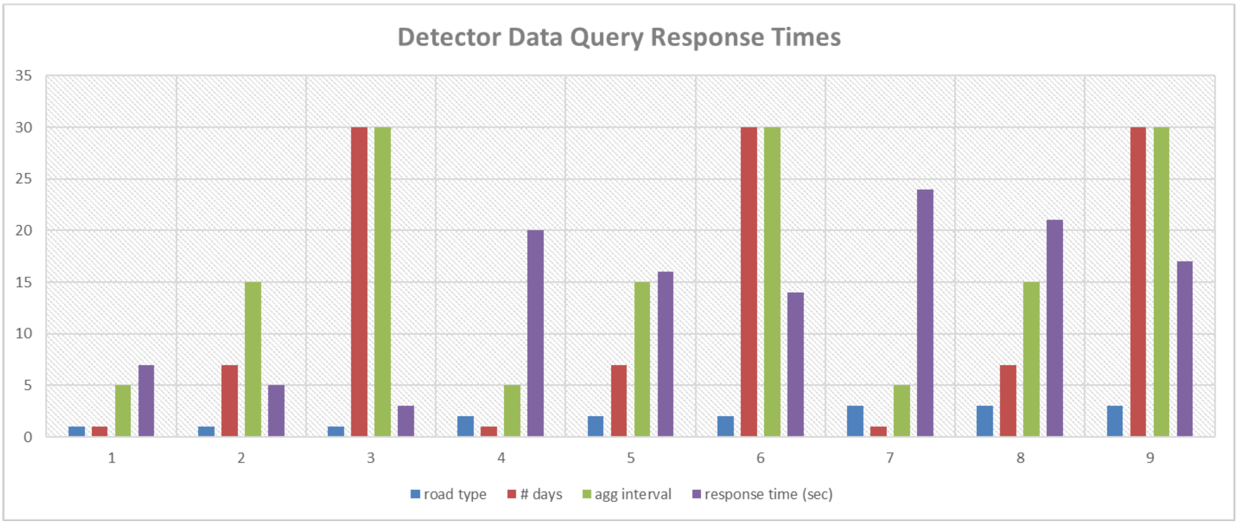


select streetname: ALL
select direction: ALL
select road type: Interstate and US Routes
Select Start Date: 2017-09-04
Select End Date: 2017-09-08
Aggregation Interval
- [✓] 5-Minutes
- [ ] 15-Minutes
- [ ] 30-Minutes



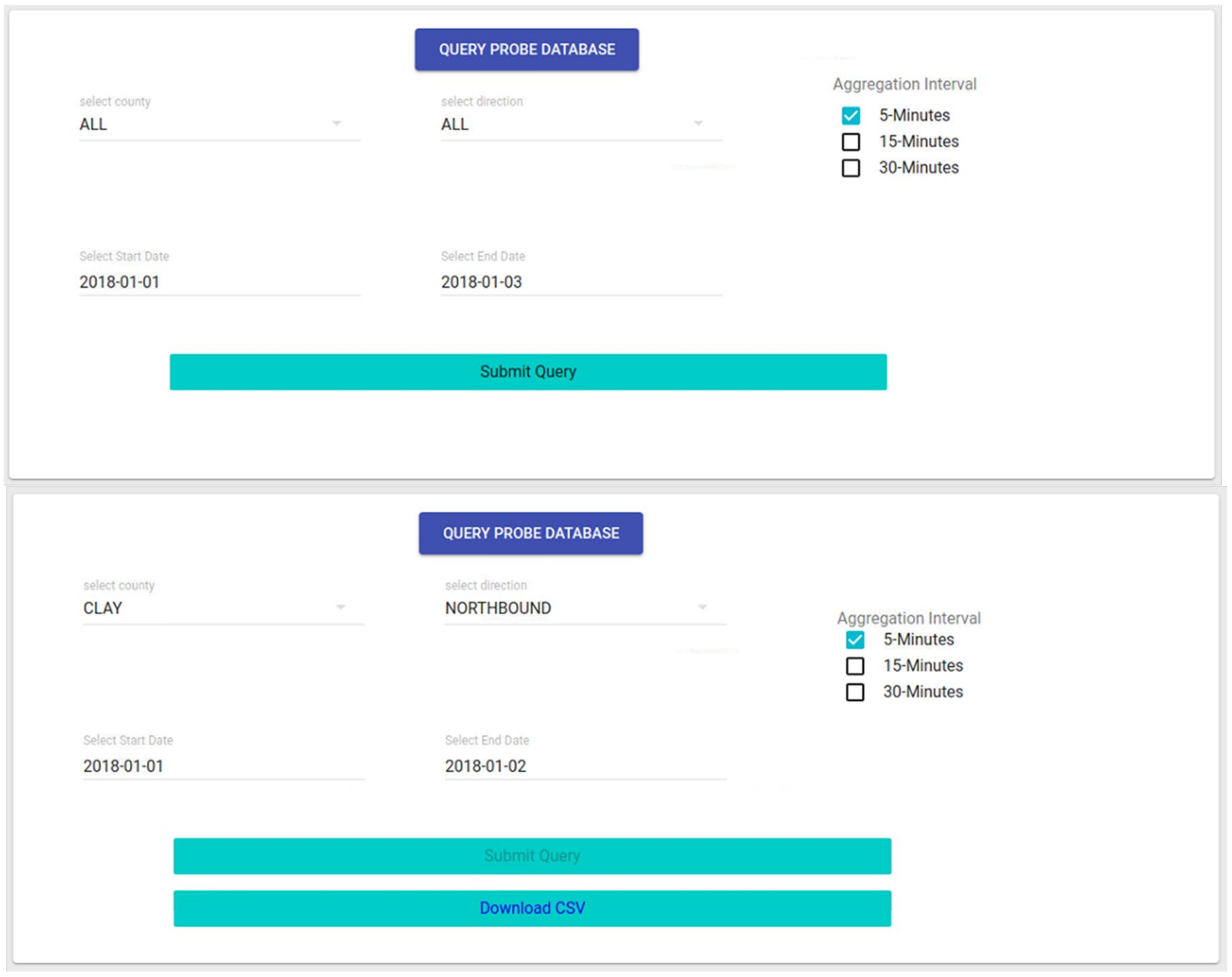
select streetname: I70
select direction: East
select road type: Interstate and US Routes
Select Start Date: 2017-09-04
Select End Date: 2017-09-08
Aggregation Interval
- [ ] 5-Minutes
- [ ] 15-Minutes
- [✓] 30-minutes

Figure 3.6. Detector Data Query Interface



	1	2	3	4	5	6	7	8	9
road type	1	1	1	2	2	2	3	3	3
# days	1	7	30	1	7	30	1	7	30
agg interval	5	15	30	5	15	30	5	15	30
response time (sec)	7	5	3	20	16	14	24	21	17

Figure 3.7. Detector Database Query Times: road types – Freeways (1), Interstates (2), all road types (3).



select county
ALL

select direction
ALL

Aggregation Interval
[✓ 5-Minutes]
15-Minutes
30-Minutes

Select Start Date
2018-01-01

Select End Date
2018-01-03

select county
CLAY

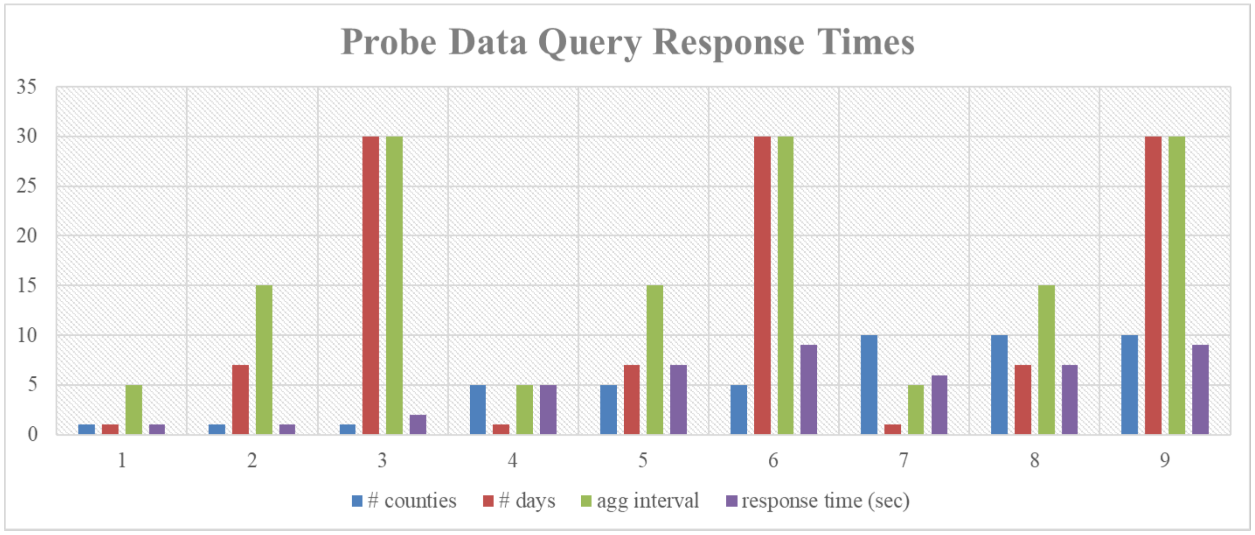
select direction
NORTHBOUND

Aggregation Interval
[✓ 5-Minutes]
15-Minutes
30-Minutes

Select Start Date
2018-01-01

Select End Date
2018-01-02

Figure 3.8. Probe Data Query Interface



	1	2	3	4	5	6	7	8	9
# counties	1	1	1	5	5	5	10	10	10
# days	1	7	30	1	7	30	1	7	30
agg interval	5	15	30	5	15	30	5	15	30
response time (sec)	1	1	2	5	7	9	6	7	9

Figure 3.9. Probe Database Query Times: Aggregation Interval in Minutes

Big Data is often defined in in terms of large versus volume, variety, and velocity. Table 3.1, shows the volume, variety and the velocity of datasets currently stored in TITAN’s data center. Statewide probe data which constitutes the largest share had about 80GB generated annually. TITAN currently has 2 years of probe data stored. At an aggregation interval of 5 minutes, a single year of detector data for St. Louis and Kansas City resulted in about 18 GB of data. Statewide crash data between 2009 and 2012 constitutes about 10% of the data stored in TITAN.

Table 3.1. Volume, Velocity and Variety of Datasets Archived on the TITAN Platform

Dataset	**Size**	**Rate**	**Period**
Probe	80GB	5 minutes	2017 - 2018
Detector	18GB	1 minute	2017 - 2018
Crash	16GB	N/A	2009 - 2012
Images	45GB	Hourly	2018
TOTAL	160 GB		

Chapter 4 Application Center – APPCENTER

The APPCENTER is the heart of TITAN. It provides a non-programmatic GUI access to underlying algorithms of the platform while guiding users to derive powerful analytical insights from their collated datasets. The design framework of APPCENTER, shown in figure 4.1, follows a big data architecture which synergistically utilizes the power of distributed computing on the server side and GPU strengths of data rendering on the client end. The simultaneous use of GPU data frames and SQL enables fast, interactive queries on the front end. Cluster resources are used for filtering, aggregating and integrating large datasets. Layout designs such as grid and list views are used to improve the user-friendliness of each application within the APPCENTER. In its current state, the APPCENTER can be used for two main activities: Performance Measurement and Predictive Analytics. Figure 4.2 shows different applications that have been developed in the app center.

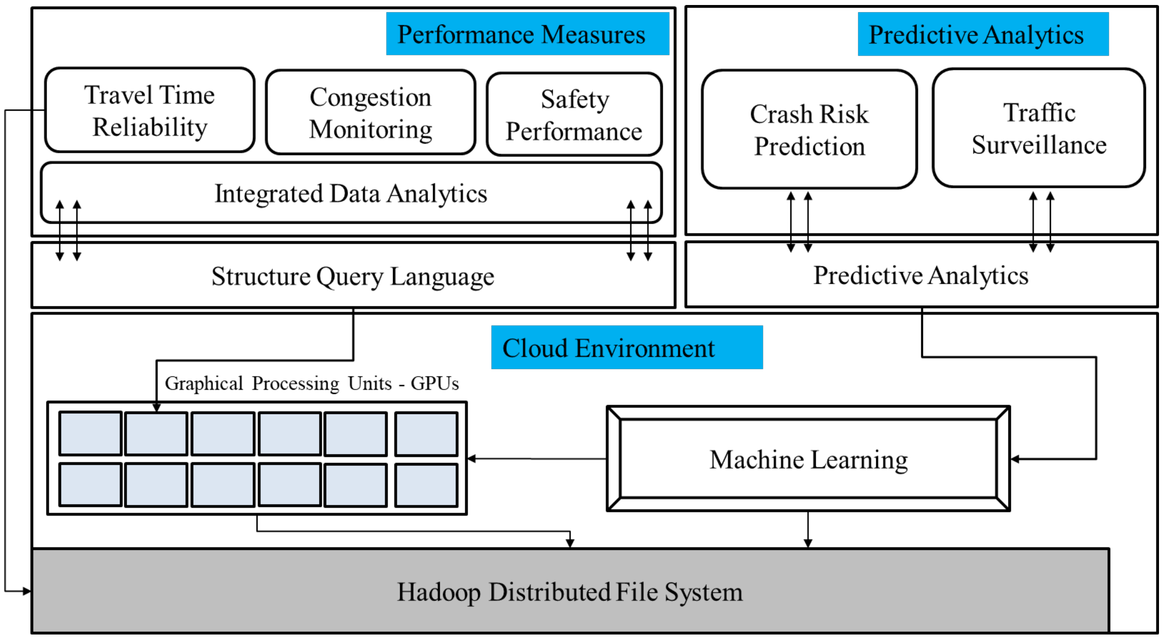


Figure 4.1. APPCENTER Design Framework

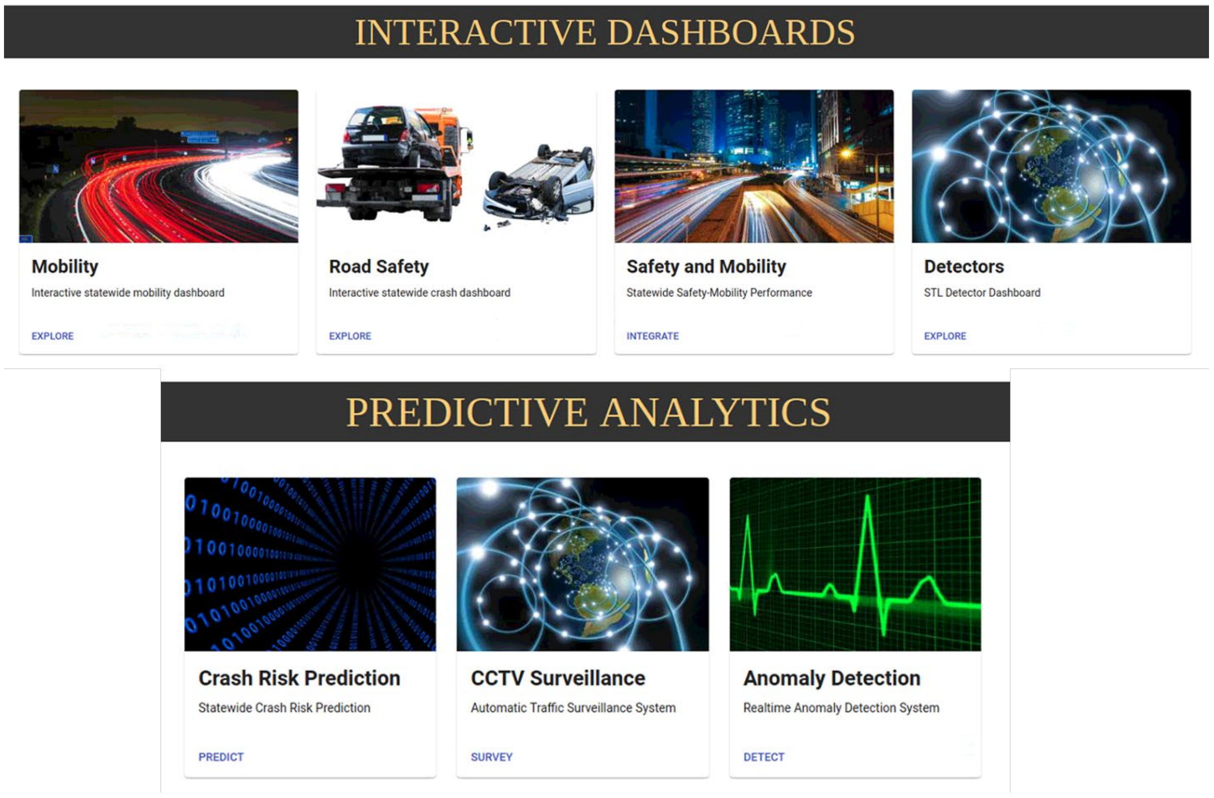

INTERACTIVE DASHBOARDS
Mobility
Interactive statewide mobility dashboard
EXPLORE
Road Safety
Interactive statewide crash dashboard
EXPLORE
Safety and Mobility
Statewide Safety-Mobility Performance
INTEGRATE
Detectors
STL Detector Dashboard
EXPLORE
PREDICTIVE ANALYTICS
Crash Risk Prediction
Statewide Crash Risk Prediction
PREDICT
CCTV Surveillance
Automatic Traffic Surveillance System
SURVEY
Anomaly Detection
Realtime Anomaly Detection System
DETECT

Figure 4.2. Applications Center - APPCENTER

4.1 Performance Measures

Transportation performance measures drive strategic investments and policy decisions that are aimed at providing safe and reliable transportation system for communities. One of the key limitations of traditional web-based methods used for communicating performance measures is their inability to provide dynamic and interactive trends at both aggregated and disaggregate levels. As a result, local, spatio-temporal trend information is usually not captured.

APPCENTER communicates a variety of performance measures via dynamic, interactive dashboards that enables users to explore both localized and generalized trends in large datasets. Users can configure routes and perform customized analysis such as travel time index, computing congestion hours, travel time reliability or speed performance, etc. APPCENTER's performance management dashboards also host functions that enable users to select, drop, pan, zoom and filter their respective areas of interest. In figure 4.3, we show an interactive dashboard example for statewide travel time reliability in 2018. This application relies on probe data for approximately 30,000 road segments in the state of Missouri. The size of the probe data generated from these segments is approximately 18GB. APPCENTER is able to visualize and perform queries on this data without latencies in the web browser.

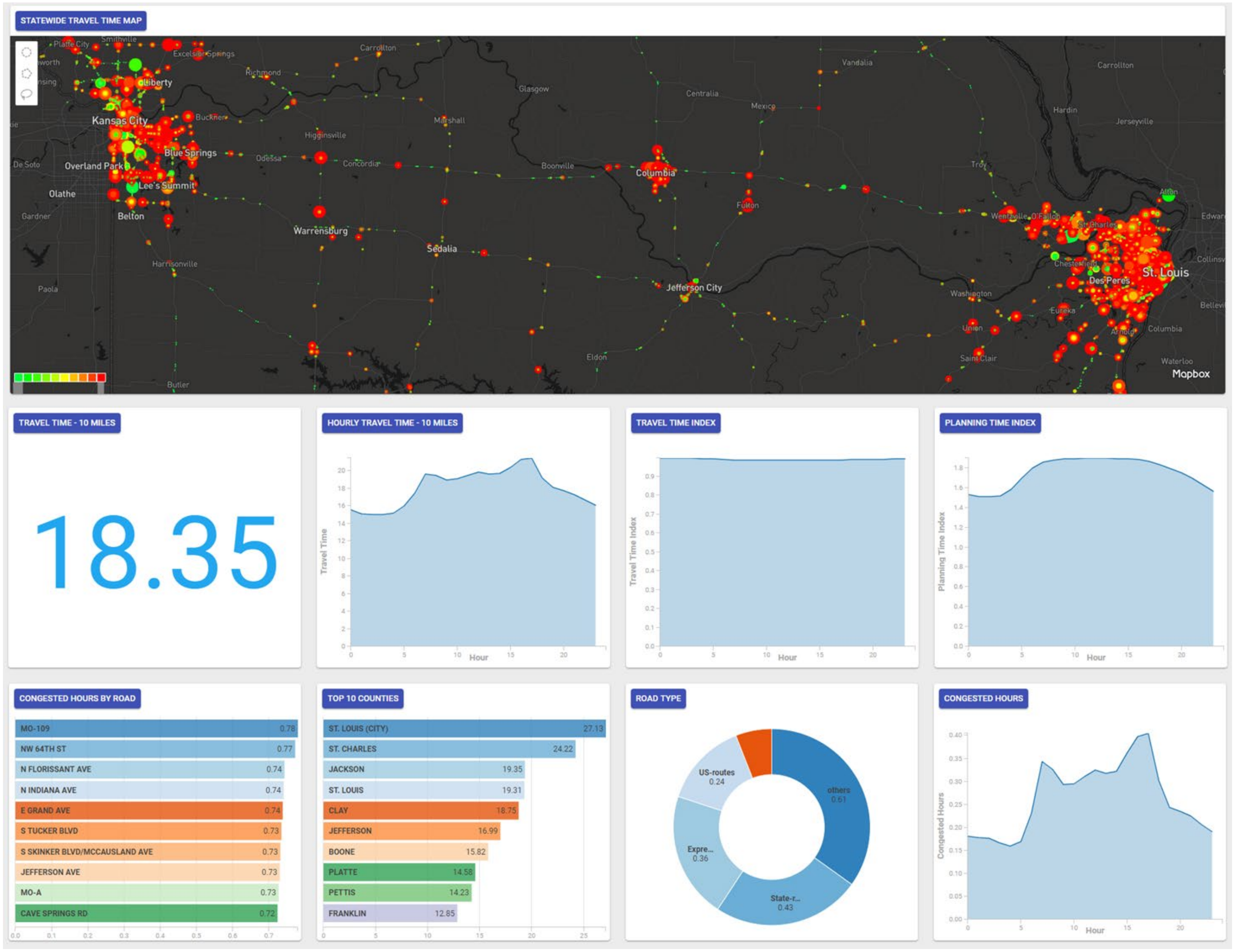


Figure 4.3. An Interactive Dashboard for Exploring Statewide Mobility Trends.

The mobility map shows the locations of all road segments with available probe data in the state of Missouri. The travel time reliability for each road segment is defined by a range of colors from red being not reliable to green being reliable. The size of the dots represents the average segment speed. The accompanying charts can be used to filter and select various areas of interest. Figure 4.4 illustrates customized functionalities that can be used to perform local analysis. The dashboard is updated to show the travel time reliability for the selected cities on the map. The analysis can also be configured for specific routes, counties, freeways, arterials, etc. Appendix A shows similar performance management applications designed for transportation safety and mobility.

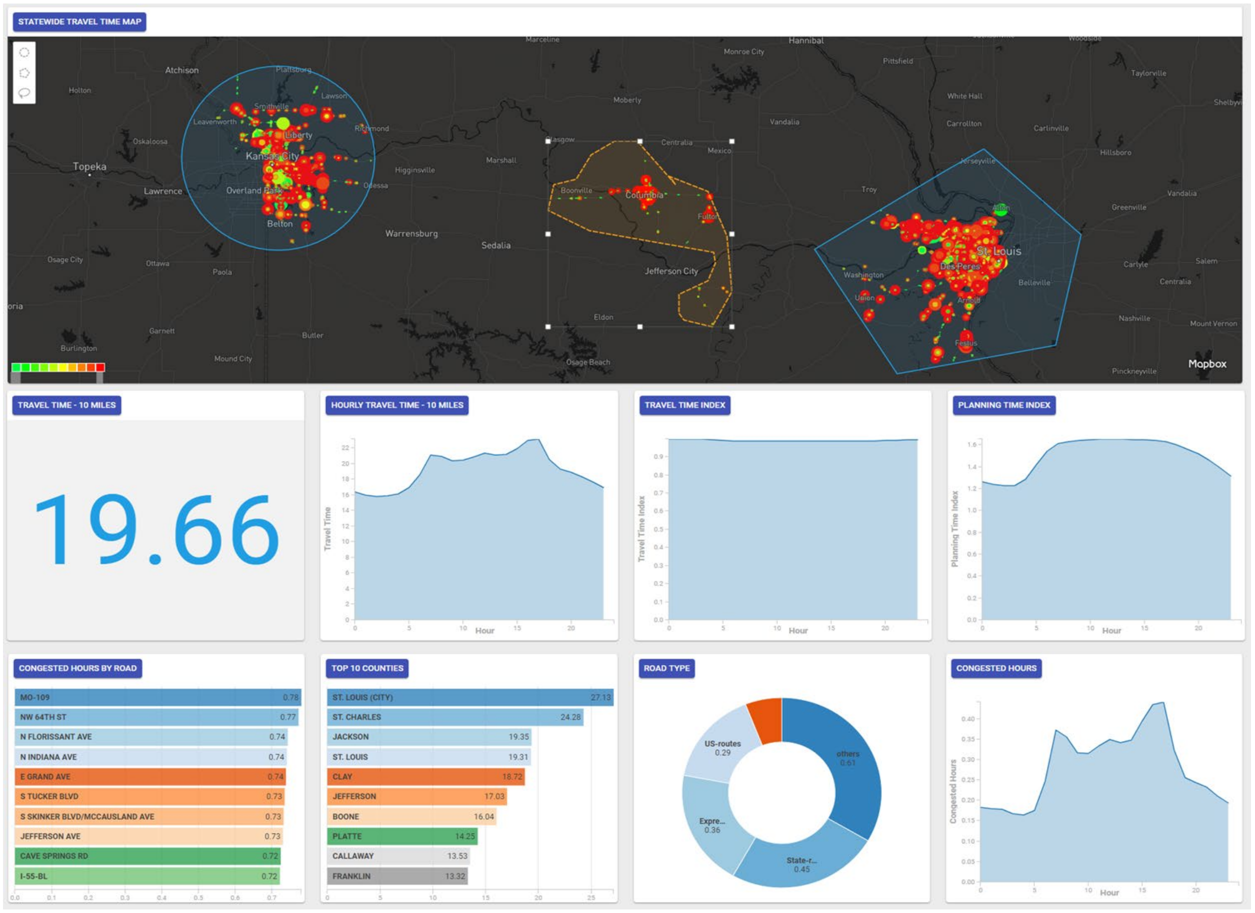


Figure 4.4. Customized Interactivity with Dashboards.

4.2 Integrated Data Analytics

The problem of data inconsistencies both in the spatial and attribute domains presents obstacles in using data for analysis, overlays and mapping. As a result, developing efficient tools for automating data harmonization and conflation has become a necessity. TITAN's data integration applications enable transportation agencies to fuse multiple data sources from disparate sources at a fast rate. The main steps for integrating multiple geographic layers are shown in figure 4.5. After the data undergoes initial pre-processing, tables can be joined by matching unique field identifiers, i.e., different databases are linked together via fields that are common to each. In cases where a unique field identifier does not exist, a GIS-based, spatial data conflation model is used to merge respective tables based on latitude/longitude.

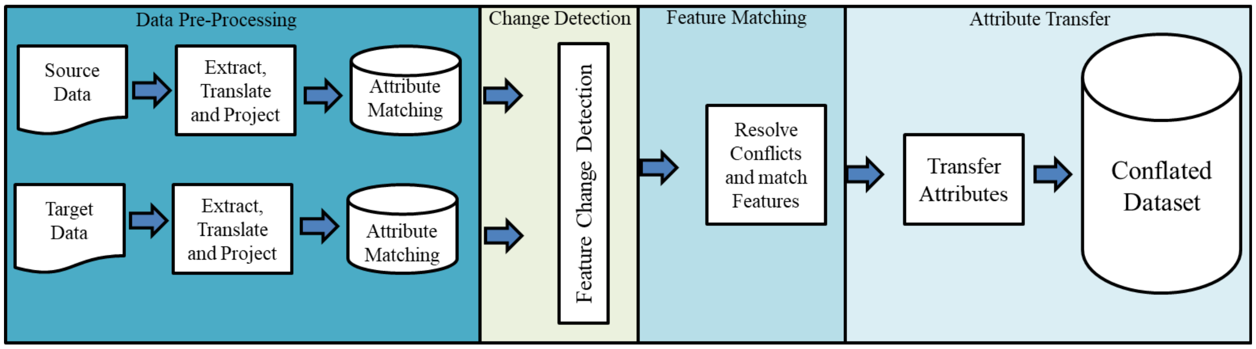


Figure 4.5. Automated Spatial Data Conflation Process.

4.2.1 Data Pre-Processing and Attribute Matching.

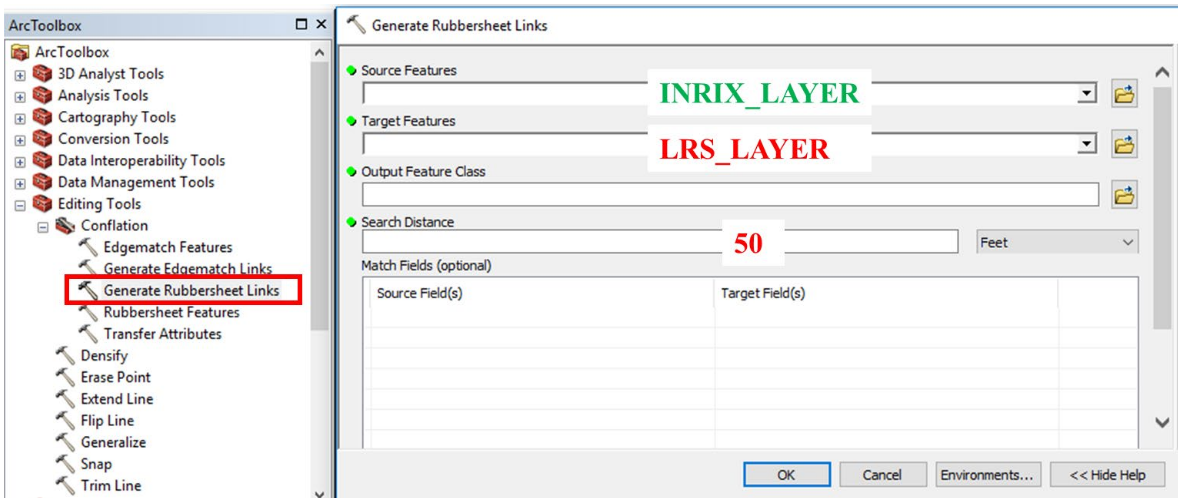
This step conditions the data for further analysis. It includes validating data geometry and topology, selecting relevant attribute features (e.g. road names, segment ids, counties, etc.) for processing and using consistent map projections to ensure that the two data layers are projected on the same geographic coordinate system. Next, attributes in both datasets describing the same features are matched. For example, some attributes that can be matched are County, Road Name and Road Directionality.

4.2.2 Detect Feature Changes

Feature change detection is the second step of the conflation process. Knowing where and what the changes are between two datasets helps to assess how significant they are and whether or not to proceed with attribute transfer. To detect feature changes in two datasets, the Detect Feature Change (DFC) tool in ESRIs ArcGIS was used. This tool identifies spatial feature differences and outputs the type of change detected for each feature. Ideally, there are four (4) possible conditions that could occur: Spatial Change (topological difference), Attribute Change, No Change (1:1 match without any spatial or attribute changes) or new feature (unmatched feature). For features where a spatial change was detected, the following workflow is used to unify and consequently conflate into the base data.

4.2.3 Feature Matching

The goal of feature matching is to map features in the source datasets which may have experienced a spatial change to its corresponding target features (base layer). ESRI's ArcGIS feature matching tools were used in this project. These tools match distorted features based on proximity, topology, pattern and similarity analysis, and other optional attributes. An output of this step is a table storing match information. The specific tool used in the current study is the "Generate Rubbersheet Tool" shown in figure 4.6. This tool generates links between matched features or points where the source and target locations are identical. In figure 4.7, the gaps in the rubbersheet results frame (green layer) shown, represents regions where roads segments were not matched. The conflation rate does fluctuate depending on the type, geometry and length of road segment.



Generate Rubbersheet Links

Source Features: INRIX_LAYER

Target Features: LRS_LAYER

Output Feature Class

Search Distance: 50 Feet

Match Fields (optional)

Source Field(s)	Target Field(s)

Figure 4.6. ArcGIS Rubbersheet Tool for Feature Matching.

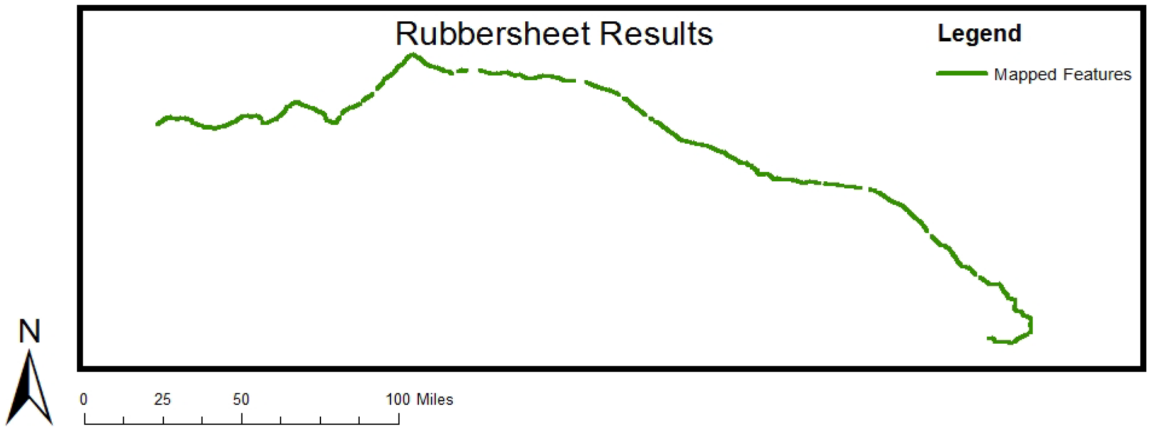


Figure 4.7. Results of Conflation. Gaps Indicate Road Sections Missed.

4.2.4 Transfer Attributes

Finally, once features between the two geographic data layers has been matched, specific feature attributes from the source layer are transferred to the matching target features. Table 2 shows an example of a transfer attribute output table for conflating statewide crash and probe data. The conflated database then allows a user to easily locate applicable data that originated from multiple databases.

Table 4.1. Transfer Attribute Output Table

TMC	ROAD	DIRECTION	MILES	ROAD ORDER	FMP	TMP	Crash ID	Crash MP
110+04890	I-64	WESTBOUND	3.960942	74	46.12782	50.08876	18005381	47.8886
110+04890	I-64	WESTBOUND	3.960942	74	46.12782	50.08876	42420941	48.9456
110+04901	I-64	WESTBOUND	2.805167	95	86.27497	89.08013	81580641	87.4706
110+04901	I-64	WESTBOUND	2.805167	95	86.27497	89.08013	81580642	87.4706
110+04901	I-64	WESTBOUND	2.805167	95	86.27497	89.08013	119635241	87.4706
110+04901	I-64	WESTBOUND	2.805167	95	86.27497	89.08013	119635242	87.4706
110+04902	I-64	WESTBOUND	5.205809	97	89.71735	94.92316	88006081	95.0636
110P04902	I-64	WESTBOUND	0.47085	98	94.92316	95.39401	88006081	95.0636

The application developed after the integration process is shown in figure 4.8. The visualization enables the user to conduct in-depth analysis of the impact of crashes on mobility and vice-versa. The map displays the locations of crashes that could be associated with road segments with available probe data between 2009 and 2012. A line plot of levels of congestion over time shows the impact of crashes on the traveling public. Other dimensions of the crash and mobility data such as congestion levels, travel time, crash severity and type can also be explored.

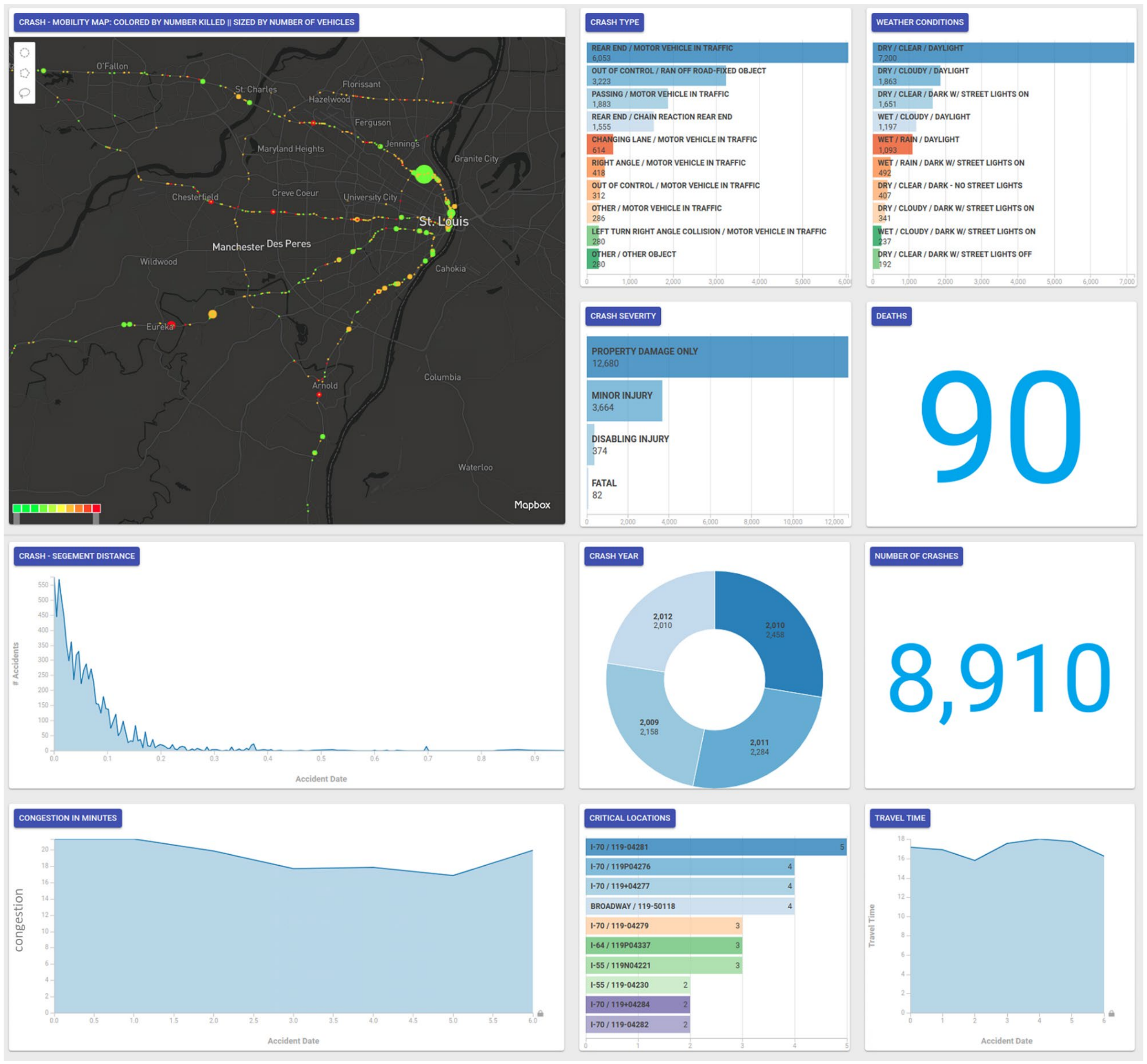

CRASH - MOBILITY MAP: COLORED BY NUMBER KILLED || SIZED BY NUMBER OF VEHICLES
O'Fallon
St. Charles
Florissant
Hazelwood
Ferguson
Maryland Heights
Jennings
Granite City
Chesterfield
Creve Coeur
University City
St. Louis
Manchester
Des Peres
Wildwood
Cahokia
Eureka
Columbia
Arnold
Waterloo
Mapbox
CRASH TYPE
REAR END / MOTOR VEHICLE IN TRAFFIC 6,053
OUT OF CONTROL / RAN OFF ROAD-FIXED OBJECT 3,223
PASSING / MOTOR VEHICLE IN TRAFFIC 1,883
REAR END / CHAIN REACTION REAR END 1,555
CHANGING LANE / MOTOR VEHICLE IN TRAFFIC 614
RIGHT ANGLE / MOTOR VEHICLE IN TRAFFIC 418
OUT OF CONTROL / MOTOR VEHICLE IN TRAFFIC 312
OTHER / MOTOR VEHICLE IN TRAFFIC 286
LEFT TURN RIGHT ANGLE COLLISION / MOTOR VEHICLE IN TRAFFIC 280
OTHER / OTHER OBJECT 280
WEATHER CONDITIONS
DRY / CLEAR / DAYLIGHT 7,200
DRY / CLOUDY / DAYLIGHT 1,863
DRY / CLEAR / DARK W/ STREET LIGHTS ON 1,651
WET / CLOUDY / DAYLIGHT 1,197
WET / RAIN / DAYLIGHT 1,093
WET / RAIN / DARK W/ STREET LIGHTS ON 492
DRY / CLEAR / DARK - NO STREET LIGHTS 407
DRY / CLOUDY / DARK W/ STREET LIGHTS ON 341
WET / CLOUDY / DARK W/ STREET LIGHTS ON 237
DRY / CLEAR / DARK W/ STREET LIGHTS OFF 192
CRASH SEVERITY
PROPERTY DAMAGE ONLY 12,680
MINOR INJURY 3,664
DISABLING INJURY 374
FATAL 82
DEATHS
90
CRASH - SEGEMENT DISTANCE
# Accidents
Accident Date
CRASH YEAR
2,012 2,010
2,010 2,458
2,009 2,158
2,011 2,284
NUMBER OF CRASHES
8,910
CONGESTION IN MINUTES
congestion
Accident Date
CRITICAL LOCATIONS
I-70 / 119-04281 5
I-70 / 119P04276 4
I-70 / 119+04277 4
BROADWAY / 119-50118 4
I-70 / 119-04279 3
I-64 / 119P04337 3
I-55 / 119N04221 3
I-55 / 119-04230 2
I-70 / 119+04284 2
I-70 / 119-04282 2
TRAVEL TIME
Travel Time
Accident Date

Figure 4.8. Statewide Safety-Mobility Dashboard

4.3 Predictive Analytics

TITAN also has the capability to learn from historical datasets and make predictions of the future. Three different predictive models were developed in the current project: (1) crash risk prediction model, (2) traffic anomaly detection model and (3) predictive model for automatic CCTV surveillance. These prototype models serve to illustrate the potential of TITAN for Big Data Analytics and are described in more detail as follows.

4.3.1 Crash Risk Prediction

The crash risk prediction model follows the national trend of leveraging a large database to improve safety decision making, much like the Highway Safety Manual (HSM). But unlike the use of the Empirical Bayes method in the HSM, TITAN uses machine learning to automate prediction using big data. The risk of a crash on a road segment within a specific time is predicted based on factors such as road segment speed differentials, weather conditions (dry, wet, snow, ice), light condition (day light, night, cloudy) and historical crash trends. The current model was trained with statewide crash, probe and weather data from 2009 through 2011 and tested on data for 2012. A visualization of predicted crashes and related accuracies are shown in figure 4.9. Due to limited training data, the models' confidence in predictions is very low especially in locations outside the major cities in Missouri. As the size of the data used to train models increases, the uncertainties for crash prediction will reduce accordingly.

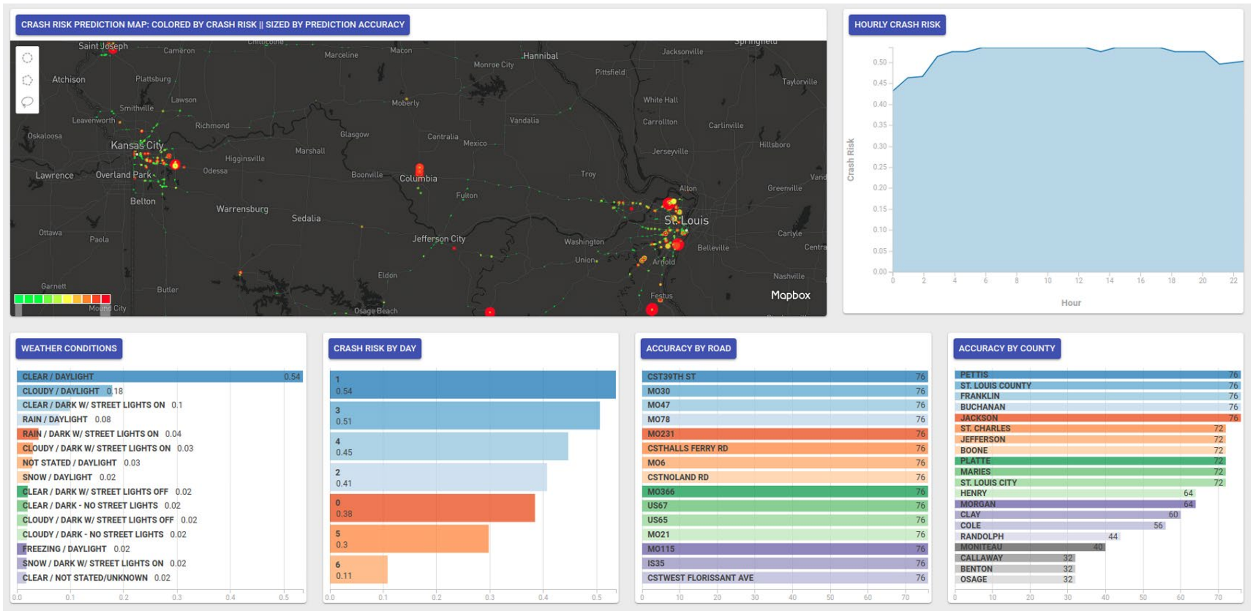


Figure 4.9. Daily Predictions More Accurate Than Hourly

4.3.2 Automated CCTV Surveillance System

Traffic surveillance is mostly manually driven. Depending on the extent of coverage, traffic management personnel are tasked to constantly monitor a wide array of cameras for events such as congestion, accidents, stranded vehicles, etc. The goal of this application is to provide a quick and automated approach for scanning CCTV cameras for traffic incidents. This will enable traffic management personnel to survey multiple cameras quickly, increasing incident detection rate and response time and reducing operator fatigue. The application is developed based on a computer vision system that is trained to detect and track various traffic incidents. This approach using computer vision defers from the traditional ways of incident detection using speeds, volumes, and occupancies. Examples of outputs from the vision system include vehicle counts, occupancy, start and end time of queues, stranded vehicle duration, and snow extent. Figure 4.10 shows a graphical user interface that is designed for the traffic management personnel to interact with the system. Users can search for different types of incidents, limit the number of cameras they want to see, and sort based on the severity or duration of incident.

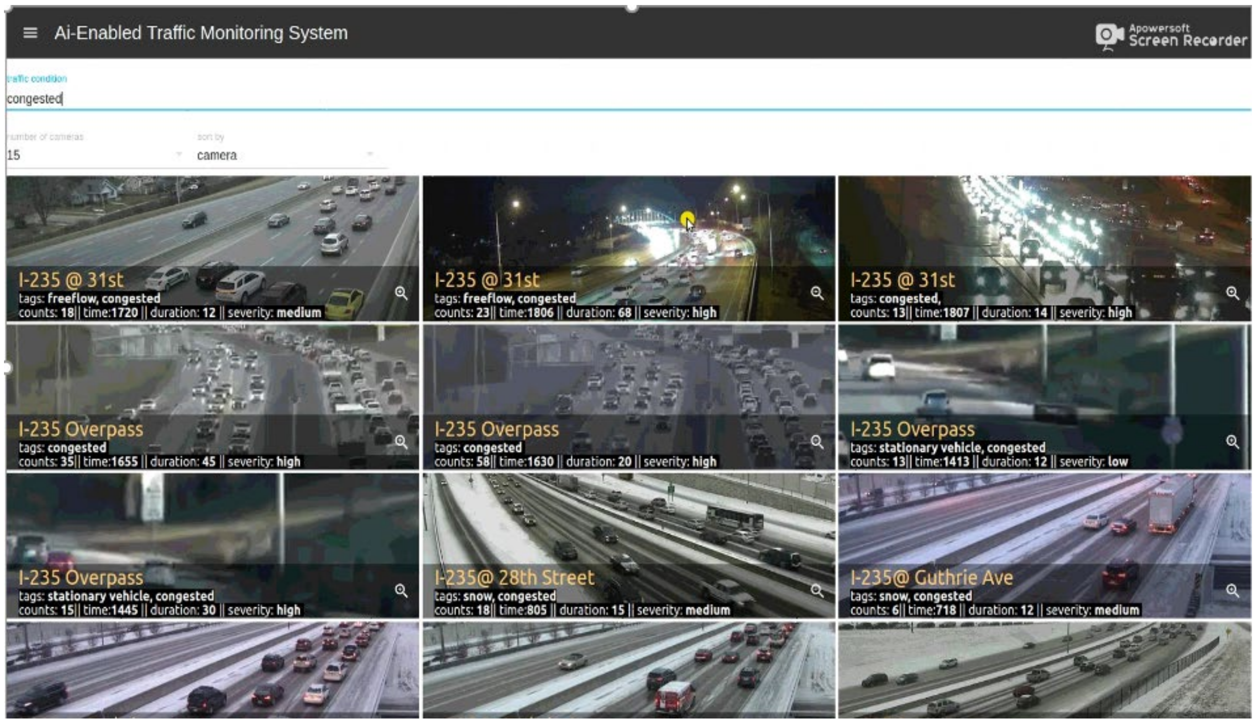

Ai-Enabled Traffic Monitoring System
Apowersoft Screen Recorder
traffic condition
congested
number of cameras
15
sort by
camera
I-235 @ 31st
tags: freeflow, congested
counts: 18|| time:1720 || duration: 12 || severity: medium
I-235 @ 31st
tags: freeflow, congested
counts: 23|| time:1806 || duration: 68 || severity: high
I-235 @ 31st
tags: congested,
counts: 13|| time:1807 || duration: 14 || severity: high
I-235 Overpass
tags: congested
counts: 35|| time:1655 || duration: 45 || severity: high
I-235 Overpass
tags: congested
counts: 58|| time:1630 || duration: 20 || severity: high
I-235 Overpass
tags: stationary vehicle, congested
counts: 13|| time:1413 || duration: 12 || severity: low
I-235 Overpass
tags: stationary vehicle, congested
counts: 15|| time:1445 || duration: 30 || severity: high
I-235@ 28th Street
tags: snow, congested
counts: 18|| time:805 || duration: 15 || severity: medium
I-235@ Guthrie Ave
tags: snow, congested
counts: 6|| time:718 || duration: 12 || severity: medium

Figure 4.10. Traffic Surveillance System: Searching for Camera with Congested Scenes, Results Sorted by Camera Name.

4.4 TITAN vs. ORACLE-Based Systems

TITAN brings together multiple databases to enable seamless integration of a variety of transportation datasets. In this section, we compare TITAN with the most common traditional relational database systems used for managing transportation data, ORACLE. We compare both platforms based on expected costs and capabilities as shown in Tables 4.2 and 4.3.

Table 4.2. Comparing TITAN Costs with Traditional Data Warehouses

Costs	**TITAN**	**Oracle**
Software Costs	Low	High
Development Costs	High	Medium Low
Cloud Deployment Costs	High	High
In-house Deployment Costs	Medium Low	Medium Low
Administration Costs	Variable	High

Table 4.3. Comparing TITAN Capabilities with Traditional Data Warehouses

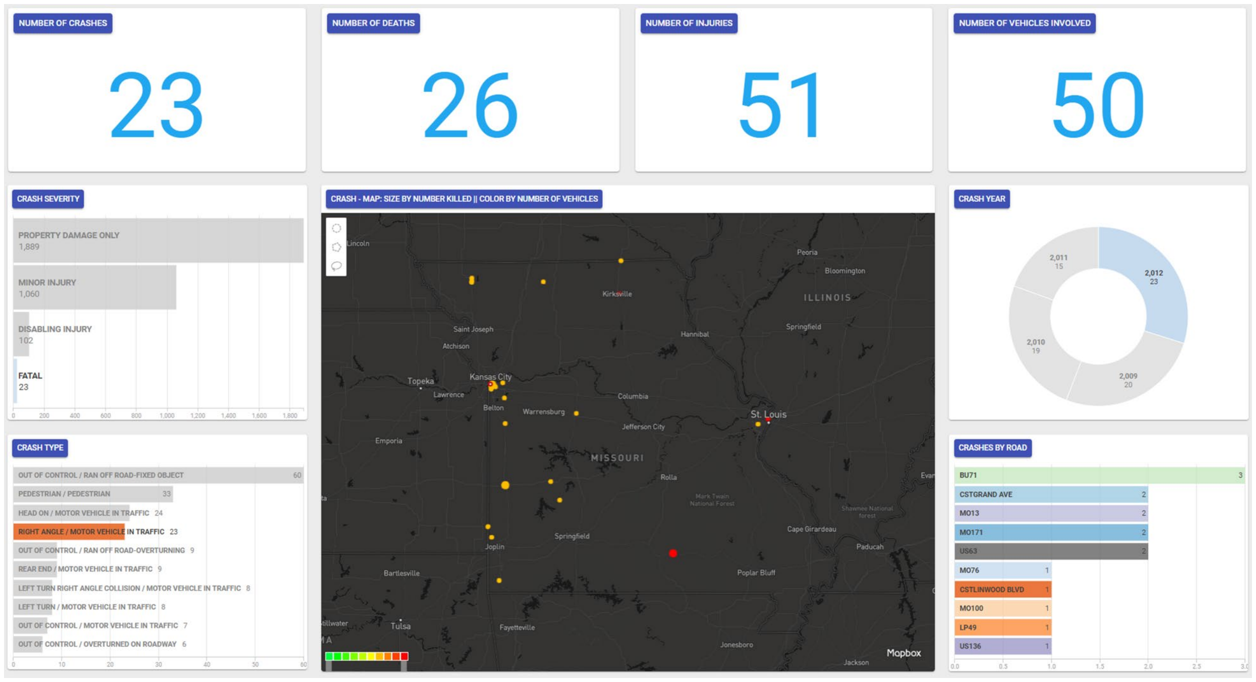
Capabilities	**TITAN**	**Oracle**
Interactive Visualizations	Yes	Yes
Data Integration	Yes	Variable
Platform Speed	High	Low
Geospatial	Yes	No
Flexibility	High	Low
Predictive Analytics	Yes	No

TITAN was developed with open-source software tools. Hence, the software costs are relatively low compared to any enterprise platform. The downside of relying heavily on open-source software is that considerable effort will have to be expended in the development of the platform. Developers will have to spend more time on integrating all the bits and pieces of code together. This increases the development cost of the TITAN compared to Oracle-based applications. Cloud costs for TITAN is expected to be slightly higher than Oracle because of the use of GPUs and computer clusters. TITAN will require at least 2 GPUs (costing about $0.5 an hour) and 5 servers in a cluster (costing about $0.35 an hour). Oracle does not need GPUs and clusters, hence will be cheaper in terms of cloud costs. Administration costs will mostly be dependent on where the platform is deployed. On the cloud, server administration is usually done by the cloud service provider. Hence, both platforms will save money. If the server is built in-house, a relational database service such as Oracle will cost significantly higher. Most of the processes and application deployed on TITAN are automated. What needs to change is the data; once the data is updated, all the apps will also be updated automatically. Hence, the role of an administrator for TITAN is significantly reduced. For Oracle, as data changes, the administrator must rewrite queries, keep track of tables, and redo models. This increases the responsibilities of the administrator, a relatively high cost.

With regards to capabilities, TITAN offers so much more. Although interactive visualizations can be carried out on Oracle, the size of data that can be visualized is limited (not more than 2GB). TITAN is able to visualize up to about 80GB of data interactively without any significant latencies on the front end. TITAN also has modules for automatically integrating data from multiple sources. Data integration on platforms such as Oracle is heavily manual driven and dependent on the size of the datasets involved. Due to the lack of GPUs and clusters, the speed of data query response, and visualizations are orders of magnitudes higher for Oracle based applications. TITAN also offers geospatial capabilities to help users deal with geographical datasets. Advanced predictive analytics on big datasets usually require the use of high computing resources such as GPUs and clusters which are all available on TITAN. Oracle based platforms are not designed for predictive analytics.

Chapter 5 Sample Applications

The following are examples of applications that utilize TITAN. These examples illustrate the capabilities of TITAN and serve to explain how TITAN could be incorporated into a person's regular work flow. Figure 5.1 shows an example of a query of fatal right-angle crashes. The type of crash, right angle, is easily queried by clicking on the particular crash type. Once the crash type is selected, all other statistics, shown in the top row, are automatically updated. Here, the top row shows there were 23 right-angle crashes, 26 fatalities resulted from those serious crashes, 51 injuries resulted, and 50 vehicles were involved.



NUMBER OF CRASHES
23
NUMBER OF DEATHS
26
NUMBER OF INJURIES
51
NUMBER OF VEHICLES INVOLVED
50
CRASH SEVERITY
PROPERTY DAMAGE ONLY 1,889
MINOR INJURY 1,060
DISABLING INJURY 102
FATAL 23
CRASH - MAP: SIZE BY NUMBER KILLED || COLOR BY NUMBER OF VEHICLES
MISSOURI
ILLINOIS
Kansas City
St. Louis
Mapbox
CRASH YEAR
2,012 23
2,011 15
2,010 19
2,009 20
CRASH TYPE
OUT OF CONTROL / RAN OFF ROAD-FIXED OBJECT 60
PEDESTRIAN / PEDESTRIAN 33
HEAD ON / MOTOR VEHICLE IN TRAFFIC 24
RIGHT ANGLE / MOTOR VEHICLE IN TRAFFIC 23
OUT OF CONTROL / RAN OFF ROAD-OVERTURNING 9
REAR END / MOTOR VEHICLE IN TRAFFIC 9
LEFT TURN RIGHT ANGLE COLLISION / MOTOR VEHICLE IN TRAFFIC 8
LEFT TURN / MOTOR VEHICLE IN TRAFFIC 8
OUT OF CONTROL / MOTOR VEHICLE IN TRAFFIC 7
OUT OF CONTROL / OVERTURNED ON ROADWAY 6
CRASHES BY ROAD
BU71 3
CSTGRAND AVE 2
MO13 2
MO171 2
US63 2
MO76 1
CSTLINWOOD BLVD 1
MO100 1
LP49 1
US136 1

Figure 5.1. Query - Number of <u>Fatalities</u> in <u>2012</u> Resulting from <u>Right Angled</u> Crashes

Figure 5.2 shows another safety example. The query is narrowed by severity (disabling injury), route (I-70), and time (2009-2012). The top row statics are once again automatically updated to reflect the narrowed selection; 173 crashes resulted in disabling injuries on I-70 from 2009 through 2012. The number of injuries (minor and disabling) resulting from the 173 crashes is 281; and 300 vehicles were involved. Any of the narrowing factors can be selected easily by clicking on the corresponding label.

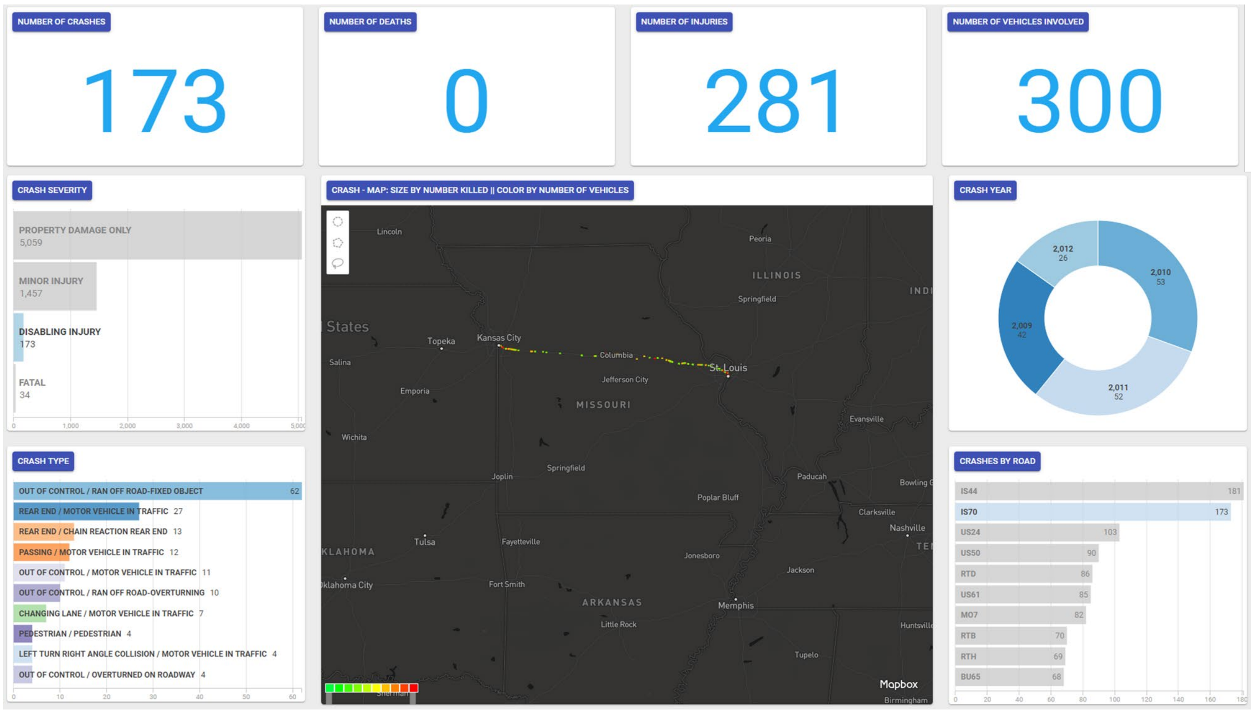

NUMBER OF CRASHES
173
NUMBER OF DEATHS
0
NUMBER OF INJURIES
281
NUMBER OF VEHICLES INVOLVED
300
CRASH SEVERITY
PROPERTY DAMAGE ONLY 5,059
MINOR INJURY 1,457
DISABLING INJURY 173
FATAL 34
CRASH - MAP: SIZE BY NUMBER KILLED || COLOR BY NUMBER OF VEHICLES
Kansas City
Columbia
Jefferson City
St. Louis
MISSOURI
ILLINOIS
Mapbox
CRASH YEAR
2,010 53
2,011 52
2,009 42
2,012 26
CRASH TYPE
OUT OF CONTROL / RAN OFF ROAD-FIXED OBJECT 62
REAR END / MOTOR VEHICLE IN TRAFFIC 27
REAR END / CHAIN REACTION REAR END 13
PASSING / MOTOR VEHICLE IN TRAFFIC 12
OUT OF CONTROL / MOTOR VEHICLE IN TRAFFIC 11
OUT OF CONTROL / RAN OFF ROAD-OVERTURNING 10
CHANGING LANE / MOTOR VEHICLE IN TRAFFIC 7
PEDESTRIAN / PEDESTRIAN 4
LEFT TURN RIGHT ANGLE COLLISION / MOTOR VEHICLE IN TRAFFIC 4
OUT OF CONTROL / OVERTURNED ON ROADWAY 4
CRASHES BY ROAD
IS44 181
IS70 173
US24 103
US50 90
RTD 86
US61 85
MO7 82
RTB 70
RTH 69
BU65 68

Figure 5.2. Query: Number of Disabling Injuries on Interstate 70 Between 2009 and 2012

Figure 5.3 illustrate an example of mobility. MoDOT often publishes regional mobility in reports such as the MoDOT Tracker. Here, the relevant mobility indices are shown for St. Louis County. The indices include average conditions, travel time reliability (e.g. planning time index), rankings of congested routes, and congestion by road type.

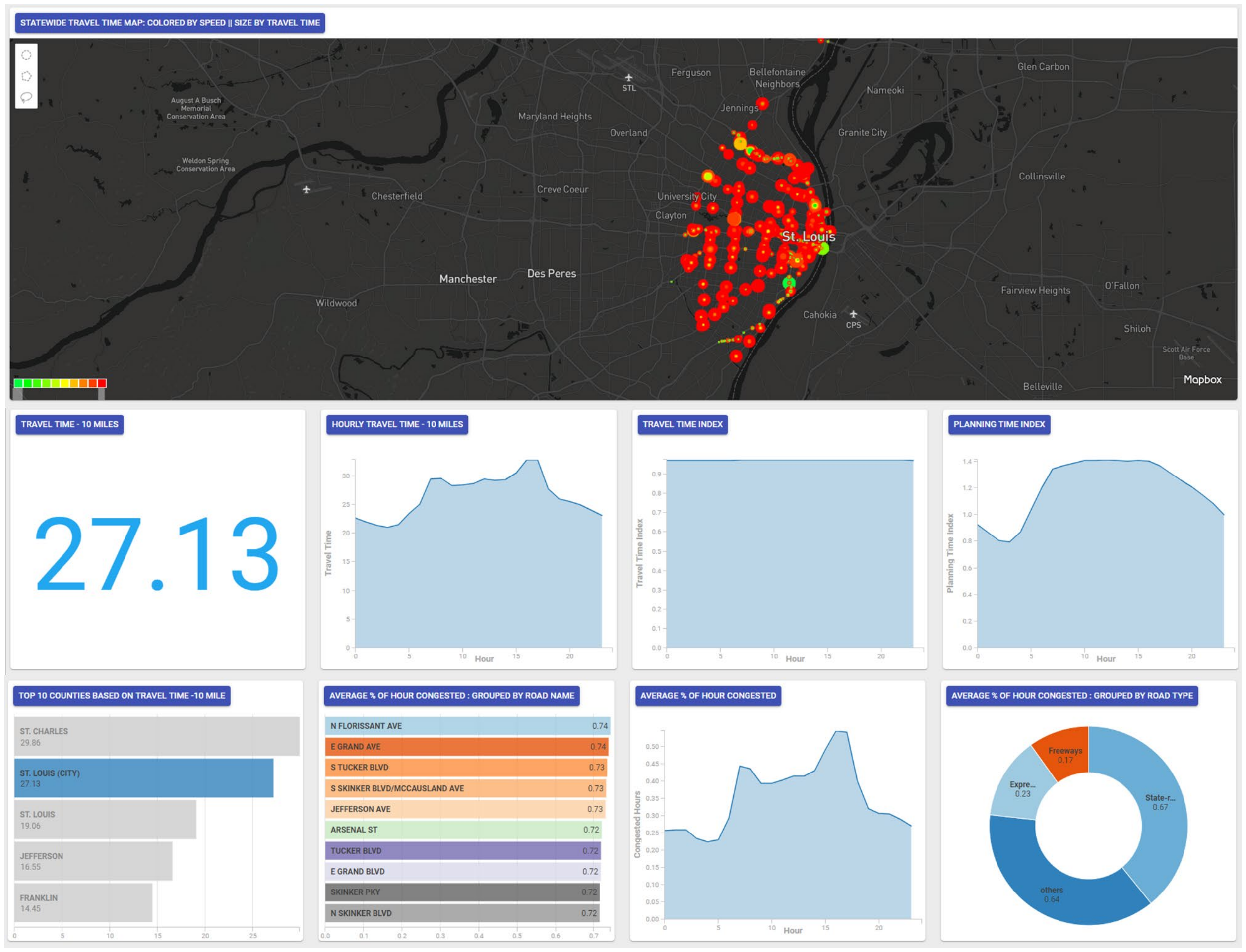


Figure 5.3. Query – Mobility Trends in St Louis County

Figure 5.4 shows how the previous query of St. Louis County could be narrowed by time or locations. For example, daytime periods of 8 am to 7 pm or only state routes.

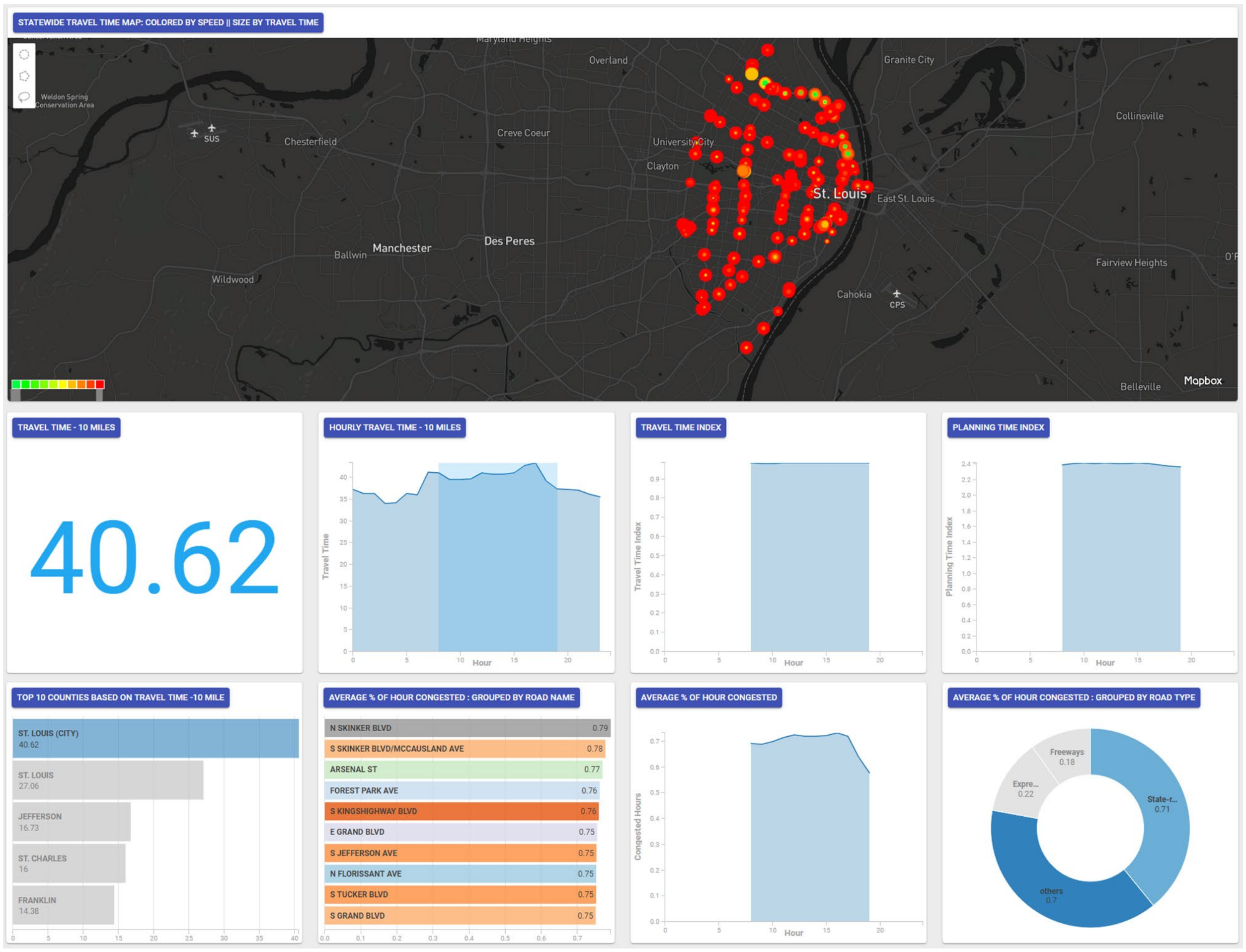


Figure 5.4. Query – Mobility Trends in St Louis County between 8am and 7pm on State Routes

Figure 5.5 shows a query in the opposite side of the state, in Jackson County, Kansas City. The query is of the am period only.

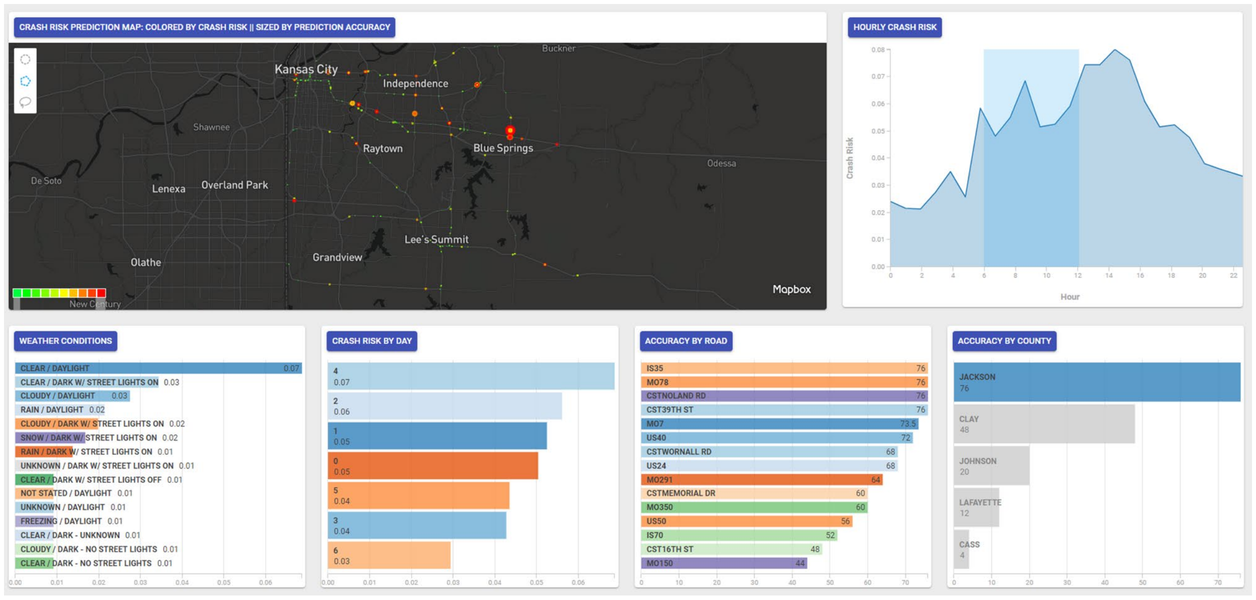


Figure 5.5 Query – Crash Prediction in Jackson County between 6am and 12pm

Figure 5.6 shows a query of multiple counties, i.e., Clay and Platte. The query also illustrates the display of weekdays only.

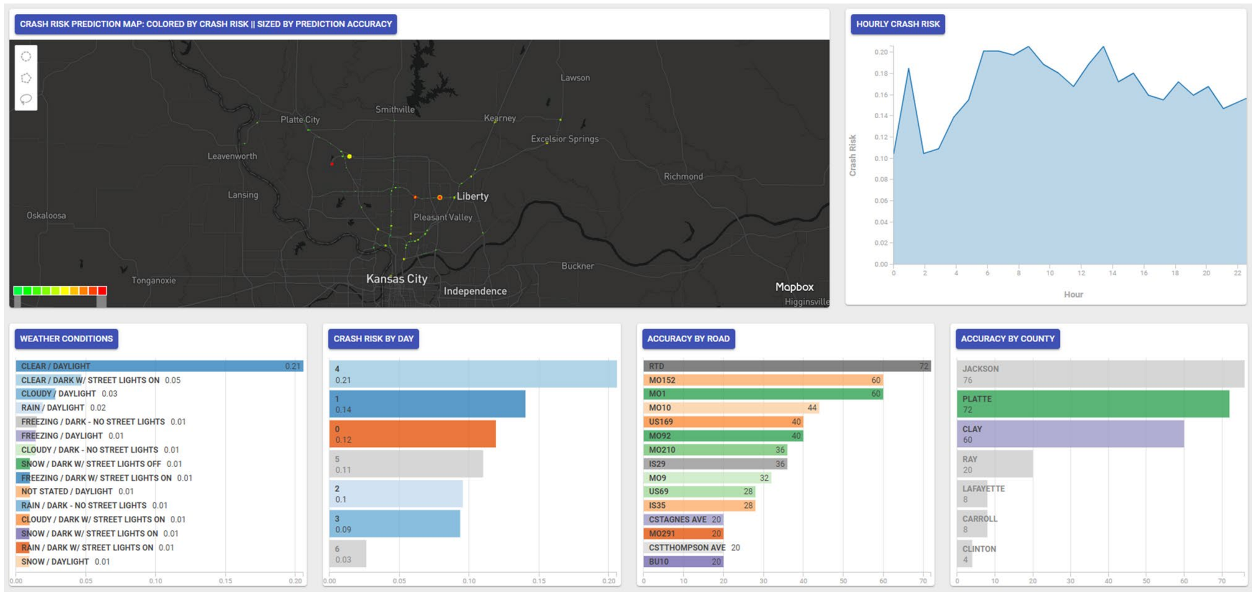


Figure 5.6. Query – Crash Prediction in Clay and Platte County during Weekdays

Figure 5.7 shows the display of traffic detector data such as speed, occupancy, and volume. All three measures are displayed graphically on top of each other so that they can be easily compared.

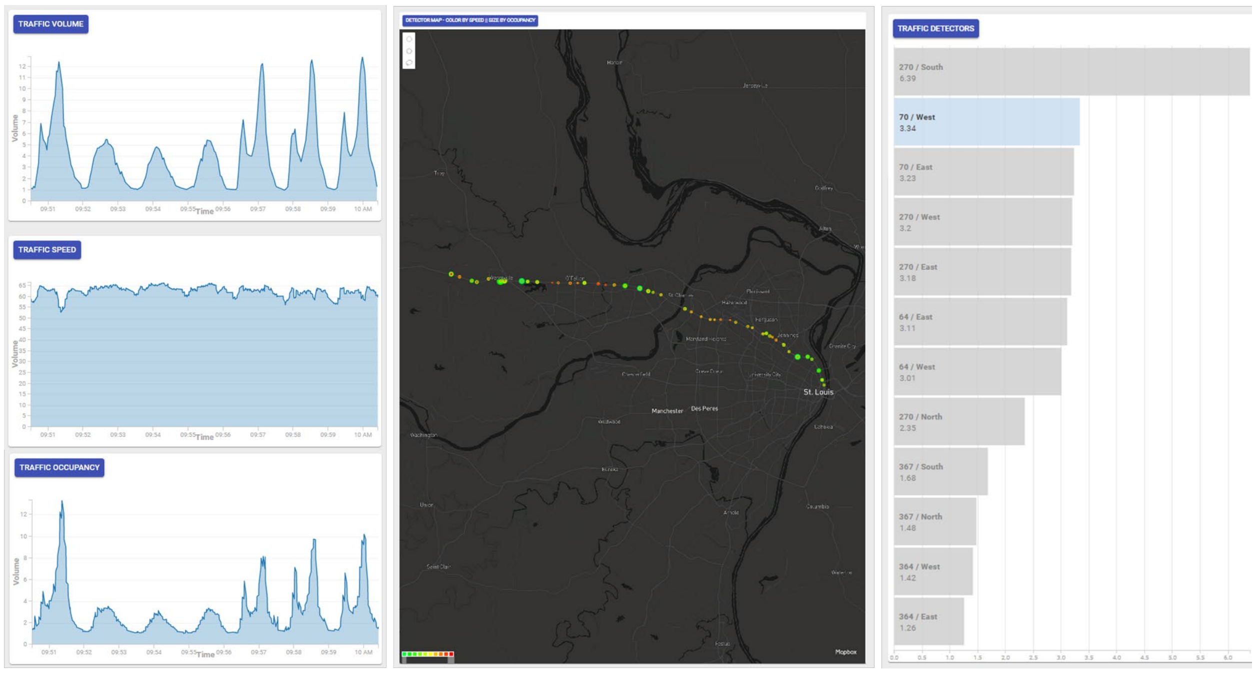
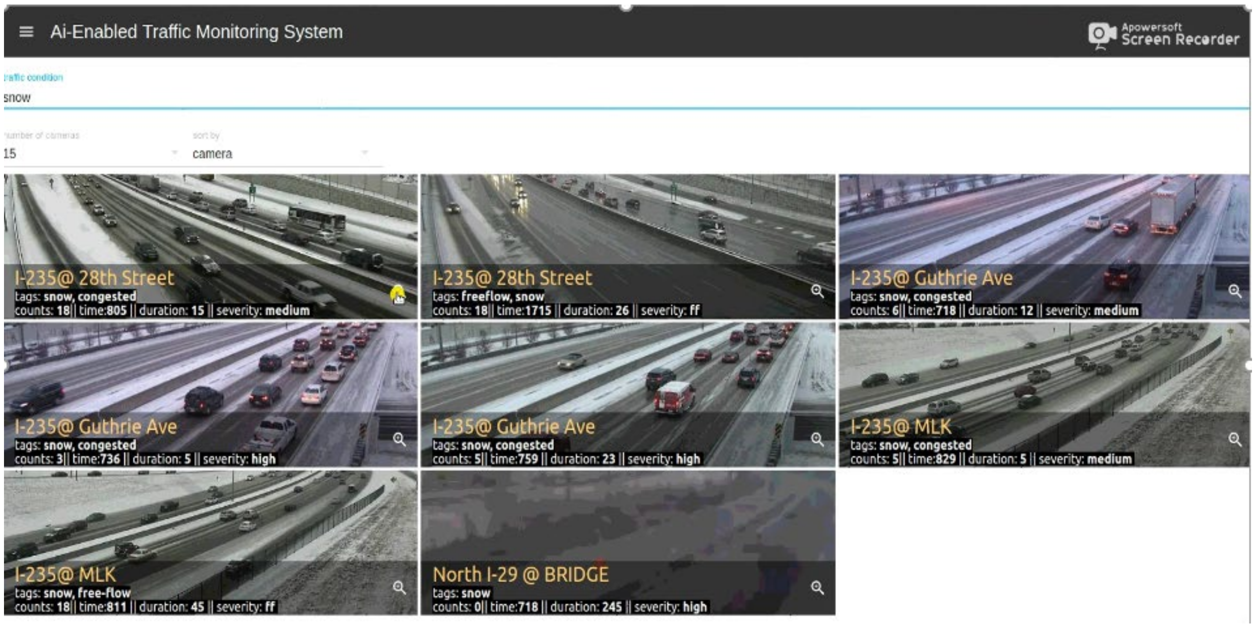


Figure 5.7. Query – Traffic <u>Speed</u>, <u>Occupancy</u> and <u>Volume</u> for <u>Interstate – 70 West</u>

Figure 5.8 illustrates the machine learning (artificial intelligence) capabilities of TITAN. All camera views were processed with machine learning, and the roads with snow accumulation were automatically displayed. Such a query could be useful in winter weather to optimize response and improve traveler information.


Ai-Enabled Traffic Monitoring System
Apowersoft Screen Recorder

raffic condition
snow

umber of cameras
15

sort by
camera

I-235@ 28th Street
tags: snow, congested
counts: 18|| time:805 || duration: 15 || severity: medium

I-235@ 28th Street
tags: freeflow, snow
counts: 18|| time:1715 || duration: 26 || severity: ff

I-235@ Guthrie Ave
tags: snow, congested
counts: 6|| time:718 || duration: 12 || severity: medium

I-235@ Guthrie Ave
tags: snow, congested
counts: 3|| time:736 || duration: 5 || severity: high

I-235@ Guthrie Ave
tags: snow, congested
counts: 5|| time:759 || duration: 23 || severity: high

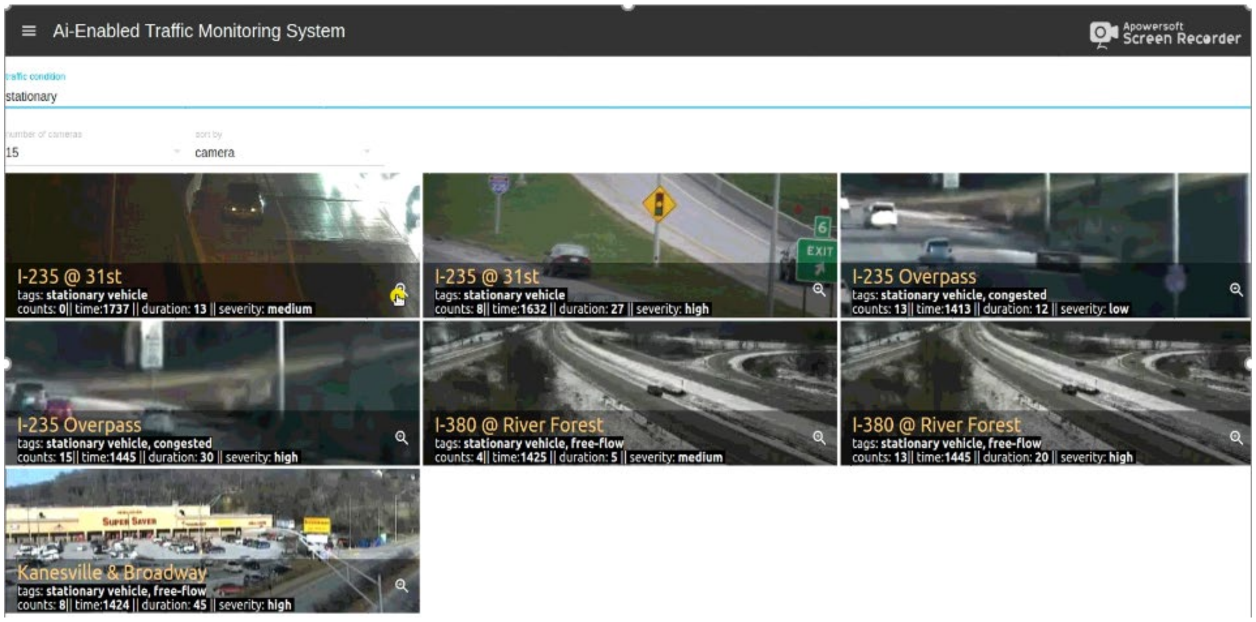
I-235@ MLK
tags: snow, congested
counts: 5|| time:829 || duration: 5 || severity: medium

I-235@ MLK
tags: snow, free-flow
counts: 18|| time:811 || duration: 45 || severity: ff

North I-29 @ BRIDGE
tags: snow
counts: 0|| time:718 || duration: 245 || severity: high

Figure 5.8. Query - Camera Locations with Snow on Ground

Figure 5.9 illustrates again TITAN's machine learning capabilities, but this time concerning incident management. Camera views were analyzed using machine learning and instances of stranded vehicles were highlighted. A quick response to such incidents can help prevent secondary crashes and to improve traffic congestion due to rubbernecking.



Ai-Enabled Traffic Monitoring System

traffic condition
stationary

number of cameras
15

sort by
camera

I-235 @ 31st
tags: stationary vehicle
counts: 0|| time:1737 || duration: 13 || severity: medium

I-235 @ 31st
tags: stationary vehicle
counts: 8|| time:1632 || duration: 27 || severity: high

I-235 Overpass
tags: stationary vehicle, congested
counts: 13|| time:1413 || duration: 12 || severity: low

I-235 Overpass
tags: stationary vehicle, congested
counts: 15|| time:1445 || duration: 30 || severity: high

I-380 @ River Forest
tags: stationary vehicle, free-flow
counts: 4|| time:1425 || duration: 5 || severity: medium

I-380 @ River Forest
tags: stationary vehicle, free-flow
counts: 13|| time:1445 || duration: 20 || severity: high

Kanesville & Broadway
tags: stationary vehicle, free-flow
counts: 8|| time:1424 || duration: 45 || severity: high

Figure 5.9. Query - Camera Locations with Stranded Vehicles

Chapter 6 Conclusion

The current project successfully designed and deployed TITAN, a fully-functional, interactive web application for storing, retrieving, integrating and visualizing a variety of large transportation datasets. By leveraging recent advances in big data, TITAN was developed with the future in mind. As data grows exponentially, TITAN scales along with it, making it an extremely fast tool for data analytics and visualization. Relying heavily on open-source tools for development resulted in a significantly cheaper, dynamic and easily customizeable platform compared to enterprise software solutions (e.g. Oracle) currently used by most transportation agencies.

A modular design approach was used to develop TITAN. It consists of a front-end and a back-end module. A user makes requests and updates by connecting to the front-end user interface. All actions from the user are subsequently passed on to the back end which sends an appropriate response back to the user on the front end. TITAN's design framework seeks to minimize the latency in communication between the front and back ends. To achieve this, data visualization on the front end was carried out by using GPUs (Graphic Processing Units), on the back end, a distributed cluster powered by Hadoop and Mongo DB.

TITAN has two main components: a data center and an applications center. The Data Center stores different streams of datasets and provides a user-friendly, non-programmatic interface for querying the different databases. By leveraging cluster computing and recent advances in big data analytics, TITAN is able to generate responses to different forms of user queries at a much faster rate compared to traditional data warehouses. The second component is APPCENTER (Applications Center) which hosts a variety of applications for performance monitoring, data integration and predictive analytics. The APPCENTER is powered with fast, interactive visualizations that enables users to identify trends and discover insights quickly for decision making. The APPCENTER was developed on top a Graphical Processing Unit (GPU) database which enables it to perform computations on large datasets within fractions of a second. Examples of applications in the APPCENTER include crash risk prediction app, safety-mobility performance measures app, traffic surveillance app, and so on.

Practically, how could MoDOT take advantage of TITAN? First, the enormous amounts of data being collected by MoDOT has great potential for improving data-driving decisions. However, such big data needs to be made accessible to various MoDOT staff who do not have the time and resources to dig into these massive databases. TITAN provides thte tools to shrink down the effort required for data integration and analysis. Second, TITAN provides graphical tools that simplifies data querying and analysis. Querying involves simple steps such as drawing a circle around a region of interest, and TITAN then automatically produces related performance measure in various formats. Third, TITAN takes advantage of modern computer cluster technology to reduce and eliminate latencies in software response. TITAN's speed is a factor in making the software user-friendly so that its use can be incorporated into the regular workflow of MoDOT employees. Future updates and developments of TITAN will include data from other areas such as freight, pavement and bridge monitoring, pedestrians and tranpsortation performance evaluation.

References

1. Adu-Gyamfi, Y., Michael, J., Skylar, K. 2016. “A Comprehensive Data Driven Evaluation of Wide Area Probe Data: Opportunities and Challenges.” Transportation Research Board, Washington, DC.
2. Chodorow, K. (2013) “MongoDB: The Definite Guide.” *O’Reilly Media Inc.*
3. Kluger, R., and Smith, B. 2013. *Next Generation Traffic Management Centers*. Final Report UVA-2012-02.
4. Mostafa Amin-Naseri, Stephen Gilbert, “Evaluating the Reliability, Coverage, and Added Value of Crowd Sourced Traffic Incident Reports from Waze,” Transportation Research Record, Washington, DC.
5. Richardson T., Gilbert S., Holub, J., Thompson, F., MacAllister, A., Radkowski, R., Winer, E., Davies, P., Terry, S. (2014) "Fusing Self-Reported and Sensor Data from Mixed-Reality Training", The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).
6. Shvachko, K., Kuang, H., Radia, S., Chansler, R. (2010) “The Hadoop Distributed File System.” *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies*, Washington DC.
7. Singh, Harmeet, Tanna, M. (2018) “Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms”, *Packt Publishing*.
8. Todd, A. (2016) “ReactJS: Become a professional in web app development,” ISBN: 1533118825 9781533118820.