

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

---

A Thesis presented to the faculty of the Graduate School  
at the University of Missouri-Columbia

---

In Partial Fulfilment  
of the Requirements for the Degree  
Master of Science in Civil Engineering

---

by  
XIAOFAN SHU  
Dr. Yaw Adu-Gyamfi, Thesis Supervisor

July 2020

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

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

Presented by Xiaofan Shu,

A candidate for the degree of Master of Science,

And hereby certify that, in their opinion, it is worthy of acceptance.

---

Dr. Yaw Adu-Gyamfi

---

Dr. Sabreena Anowar

---

Dr. Timothy C. Matisziw

## **ACKNOWLEDGMENTS**

Firstly, I would like to thank my advisor Prof. Yaw Adu-Gyamfi who always provides me help when I need him especially on my research and courses. He is super busy but when his students need him, he is always there. What is more, Dr. Yaw is very skillful and professional in transportation field, so he can give his students very professional guide and suggestions on learning and research. I really appreciate him.

Then, I would like to thank my thesis committee members: Prof. Timothy C. Matisziw and Prof. Sabreena Anowar for their time, guide and feedbacks on my thesis.

Then, I would like to thank my parents who always support me and accompany me. Another person I want to thank is my friend Peng Jin. He offers me a lot of help on my life and research. Thanks for his supporting and accompanying. Finally, I would like to thank all professors and colleagues in transportation lab. We are just like a big family.

# TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS .....</b>	<b>II</b>
<b>TABLE OF CONTENTS .....</b>	<b>III</b>
<b>LIST OF FIGURES.....</b>	<b>V</b>
<b>LIST OF TABLES.....</b>	<b>VII</b>
<b>ABSTRACT .....</b>	<b>VIII</b>
<b>CHAPTER 1: INTRODUCTION .....</b>	<b>1</b>
1.1 DATA INTEGRATION AND ANALYTICS .....	2
1.1.1 Data Integration.....	2
1.1.2 Big Data Analytics.....	3
1.2 PROBLEM STATEMENT.....	4
1.3 OBJECTIVES OF THE STUDY.....	5
<b>CHAPTER 2: LITERATURE REVIEW.....</b>	<b>7</b>
<b>CHAPTER 3: PROPOSED METHODOLOGY .....</b>	<b>12</b>
3.1 TITAN DESIGN APPROACH .....	12
3.1.1 Back-End Design.....	13
3.1.1.1 Firebase DB .....	14
3.1.1.2 Mongo DB .....	15
3.1.1.3 Graphical Processing Unit (GPU) DB.....	16
3.1.2 Data Integration Layer .....	16
3.1.2.1 Data Pre-Processing and Attribute Matching .....	17
3.1.2.2 Detect Feature Changes .....	17
3.1.2.3 Feature Matching.....	18
3.1.2.4 Transfer Attributes .....	19
3.1.3 Conflation.....	20
3.1.3.1 Link Four Types of Data.....	21
3.1.3.2 Three Types of Conflation.....	22
3.1.3.2.1 Conflate Point to Line .....	22
3.1.3.2.2 Conflate Line to Line .....	22
3.1.3.2.3 Conflate Milepost to Line.....	24
3.1.4 Front-End Design .....	25
3.1.5 Data Center.....	26
3.1.4.1 Data Upload .....	27
3.1.4.2 Data Query .....	28
3.2 TITAN APPLICATION.....	35
3.2.1 Mobility.....	37
3.2.2 Transit .....	39
3.2.3 Safety and Mobility .....	40
3.2.4 Prediction Analysis.....	41
3.2.4.1 Crash Risk Prediction .....	42

3.2.4.2 Automated CCTV Surveillance System .....	43
<b>CHAPTER 4: PERFORMANCE EVALUATION.....</b>	<b>45</b>
4.1 COMPARE WITH ORACLE .....	45
4.2 COMPARE WITH TABLEAU .....	47
<b>CHAPTER 5: SAMPLE APPLICATIONS .....</b>	<b>50</b>
<b>CHAPTER 6: CONCLUSION AND FUTURE RESEARCH.....</b>	<b>57</b>
<b>REFERENCES .....</b>	<b>59</b>

## LIST OF FIGURES

<b>FIGURE 1. SCHEMATIC OF TITAN’S KEY COMPONENTS .....</b>	<b>12</b>
<b>FIGURE 2. BACK-END DESIGN FRAMEWORK .....</b>	<b>14</b>
<b>FIGURE 3. DATA STORAGE IN MONGO DB .....</b>	<b>15</b>
<b>FIGURE 4. AUTOMATED SPATIAL DATA CONFLATION PROCESS.....</b>	<b>17</b>
<b>FIGURE 5. ARCGIS RUBBERSHEET TOOL AND RESULTS OF CONFLATION. GAPS</b>	
<b>INDICATE ROAD SECTIONS MISSED .....</b>	<b>19</b>
<b>FIGURE 6. ARCMAP OF PAIRED AND UNPAIRED SEGMENTS .....</b>	<b>24</b>
<b>FIGURE 7. DESIGN COMPONENTS OF DATA CENTER.....</b>	<b>26</b>
<b>FIGURE 8. DATA CENTER.....</b>	<b>27</b>
<b>FIGURE 9. UPLOADING DATA TO TITAN .....</b>	<b>28</b>
<b>FIGURE 10. CRASH DATA QUERY INTERFACE.....</b>	<b>29</b>
<b>FIGURE 11. CRASH DATABASE QUERY TIMES: ROAD TYPES – FREEWAYS (1),</b>	
<b>INTERSTATES (2), ALL ROAD TYPES (3). ACCIDENT SEVERITY – FATAL (1),</b>	
<b>DISABLING INJURY (2), MINOR INJURY (3), PROPERTY DAMAGE (4).....</b>	<b>30</b>
<b>FIGURE 12. DETECTOR DATA QUERY INTERFACE .....</b>	<b>31</b>
<b>FIGURE 13. DETECTOR DATABASE QUERY TIMES: ROAD TYPES – FREEWAYS (1),</b>	
<b>INTERSTATES (2), ALL ROAD TYPES (3).....</b>	<b>32</b>
<b>FIGURE 14. PROBE DATA QUERY INTERFACE.....</b>	<b>33</b>
<b>FIGURE 15. PROBE DATABASE QUERY TIMES: AGGREGATION INTERVAL IN MINUTES.....</b>	<b>34</b>
<b>FIGURE 16. APPCENTER DESIGN FRAMEWORK .....</b>	<b>36</b>
<b>FIGURE 17. APPLICATIONS CENTER – APPCENTER.....</b>	<b>36</b>
<b>FIGURE 18. AN INTERACTIVE DASHBOARD FOR EXPLORING STATEWIDE MOBILITY</b>	

TRENDS .....	38
FIGURE 19. AN INTERACTIVE DASHBOARD FOR EXPLORING TRANSIT DASHBOARD .....	40
FIGURE 20. STATEWIDE SAFETY-MOBILITY DASHBOARD .....	41
FIGURE 21. DAILY PREDICTIONS MORE ACCURATE THAN HOURLY.....	43
FIGURE 22. TRAFFIC SURVEILLANCE SYSTEM: SEARCHING FOR CAMERA WITH CONGESTED SCENES, RESULTS SORTED BY CAMERA NAME.....	44
FIGURE 23. DATA VISUALIZATION BY TABLEAU .....	48
FIGURE 24. QUERY – NUMBER OF FATALITIES IN 2012 RESULTING FROM RIGHT- ANGLED CRASHES.....	50
FIGURE 25. QUERY – NUMBER OF DISABLING INJURIES ON INTERSTATE 70 BETWEEN 2009 AND 2012 .....	51
FIGURE 26. QUERY – MOBILITY TRENDS IN ST LOUIS COUNTY .....	52
FIGURE 27. QUERY – MOBILITY TRENDS IN ST LOUIS COUNTY BETWEEN 8 AM AND 7 PM ON STATE ROUTES.....	53
FIGURE 28. QUERY – CRASH PREDICTION IN JACKSON COUNTY BETWEEN 6 AM AND 12 PM .....	54
FIGURE 29. QUERY – CRASH PREDICTION IN CLAY AND PLATTE COUNTY DURING WEEKDAY .....	54
FIGURE 30. QUERY – TRAFFIC SPEED, OCCUPANCY AND VOLUME FOR INTERSTATE – 70 WEST .....	55
FIGURE 31. QUERY - CAMERA LOCATIONS WITH SNOW ON GROUND.....	56
FIGURE 32. QUERY - CAMERA LOCATIONS WITH STRANDED VEHICLES.....	56

## LIST OF TABLES

<b>TABLE 1. TRANSFER ATTRIBUTE OUTPUT TABLE.....</b>	<b>20</b>
<b>TABLE 2. CHOOSING A FRONT-END LIBRARY FOR TITAN’S DEVELOPMENT.....</b>	<b>25</b>
<b>TABLE 3. VOLUME, VELOCITY, AND VARIETY OF DATASETS ARCHIVED ON THE TITAN PLATFORM.....</b>	<b>37</b>
<b>TABLE 4. COMPARING TITAN WITH TRADITIONAL DATA WAREHOUSES.....</b>	<b>45</b>
<b>TABLE 5. COMPARISON OF MAPD AND TABLEAU .....</b>	<b>47</b>
<b>TABLE 6. COMPARING TITAN WITH TABLEAU .....</b>	<b>48</b>



## **ABSTRACT**

State transportation agencies regularly collect and store 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 require the use of advanced tools such as big data analytics, machine learning, and cluster computing. TITAN (Transportation data InTegration and ANalytics) is an initial prototype of an interactive web-based platform that demonstrates the possibilities of such big data software. The current study succeeded in showing a user-friendly front end, graphical in nature, and a scalable back end capable of integrating multiple big databases with minimal latencies. This thesis documents how the key components of TITAN were designed. Several applications, including mobility, safety, transit performance, and predictive crash analytics, are used to explore the strengths and limitations of the platform. A comparative analysis of the current TITAN platform with traditional database systems such as Oracle and Tableau is also conducted to explain who needs to use the platform and when to use which platform. As TITAN was shown to be feasible and efficient, the future research direction should aim to add more types of data and deploy TITAN in various data-driven decision-making processes.

# CHAPTER 1: INTRODUCTION

In recent years, with the development of technology and rapid popularization and application of the Internet, we have entered the era of big data. The development of the Internet has brought us great convenience. Currently, all aspects of our lives are inextricably linked to the Internet. We are now in an era of data explosion. Various types of data are flooding our lives, and the widespread use of the Internet combines them to form big data. A more optimized system is needed to analyze, process, and store such big data. Big data analytics refer to a new model that can quickly organize and process a large amount of information and data in a short time. It surpasses the earlier backward model, which is a more optimized data processing model. Data adds immense convenience to our lives. The Internet has entered various aspects of our lives in a step-by-step manner. Our study, work, and life are all served by the Internet.

Furthermore, the temporality and convenience brought by big data are accompanied by more challenges. Moreover, under the influence of big data, the transportation system has been greatly affected. It is becoming increasingly larger and more complex, meaning that the information and data in the transportation system must be processed more than before and in a better and faster manner. In the past, the processing speed of traffic information was very slow, requiring more manual integration and analytics and excessive time. Additionally, manual processing has a serious disadvantage; it is highly error-prone, and the error rate is very high. At this time, the advantages of big data are reflected. Big data of transportation can be integrated and analyzed by a machine that can quickly draw conclusions.

Big data integration and analytics is an advanced analytic technique that handles

large and diverse datasets. Big data has three common characteristics called 3Vs: high volume, high variety, and high variety. As big data has very high volume, complex data structure, and various data sources, an effective platform for data integration and analytics is necessary. In the transportation field, big data analytics techniques also keep developing.

Recent advances in big data analytics are enabling organizations to digest humongous amounts of data and transform them into actionable insights (Adu-Gyamfi et al. 2016; Kluger and Smith 2013). This innovation is being fueled by massive open data platforms, driven by machine-learning and empowered by low-cost cloud computing. The open data platform is primarily designed to create Hadoop-powered big data applications on a common platform, which provides big data developers a basic model to build applications and services that can be interoperable on different platforms. This new wave of invention could be leveraged to enable transportation agencies to identify the usefulness of their diverse datasets and explore previously untapped applications.

## **1.1 Data Integration and Analytics**

### **1.1.1 Data Integration**

Big data integration is the process of conflating different sources and format data into a single, unified view. Due to different development times or departments, there are often multiple heterogeneous information systems running simultaneously on different software and hardware platforms. The data sources of these systems are independent and closed from each other, making it difficult to store data in communication, sharing, and integration phases between the systems; therefore, an “information isolated island” is formed. With the continuous in-depth application of information technology, it is urgent to integrate the existing information and connect with “information isolated islands” to

share information. In recent decades, with the rapid development of science and technology, human society has gathered a huge amount of data; moreover, the instances of data collection, storage, processing, and dissemination have increased. Data sharing can enable more people to make fuller use of existing data resources and reduce repeated labor and corresponding costs in data collection. However, in the process of implementing data sharing, as the data provided by different users may come from different paths and sources, the data content, format, and quality are very different. Furthermore, the data format cannot be conflated, or the information is lost after data conflation, which seriously damages the flow and sharing of data in various departments and software systems. Therefore, how to effectively integrate data management is an urgent issue that needs to be solved. Nowadays, big data integration still faces challenges such as uncertainty of data, syncing across data sources, slow integration speed, etc.

### **1.1.2 Big Data Analytics**

Big data is a collection of data that cannot be captured, managed, and processed with regular software tools on a commodity computer in a reasonable time period. The process of collecting, organizing, visualizing, and analyzing large size datasets to get more hidden and useful information from data is called big data analytics. There are five common big data analytic methods. The first is visualization analytics. Users of big data analytics include experts and ordinary users, but its most basic requirement is visualization analytics because they can intuitively present the characteristics of big data and be easily accepted by the audience, which is just like watching simple pictures. The second method is data mining algorithms, which is the core of the big data analytics theory. Various data mining algorithms can scientifically present the characteristics of the

data itself according to different data types and formats. On the other hand, data mining algorithms can process big data faster. If an algorithm takes years to get results, the value of big data cannot be shown. The third way is predictive analytics, which is one of the ultimate application areas of big data analytics. Mining features from big data include building a model and then introducing new data through the model to predict future data. The fourth method is the semantic engine. The diversification of unstructured data brings new challenges to data analytics. A set of tools is needed to analyze and improve the data. Enough artificial intelligence is needed to design the semantic engine to proactively extract information from the data. The last approach is data quality and management. Big data analytics is inextricably linked to data quality and management. Whether in academic research or commercial applications, high-quality data and effective data management can ensure the authenticity and value of analysis results. The aforementioned five aspects form the basis of big data analytics. For deeper big data analysis, more professional and deeper methods are available. This research mainly focuses on two methods: visualization analytics and predictive analytics.

## **1.2 Problem Statement**

The recent surge in the use of community-based sensing such as Waze, ubiquitous mobile computing, automatic vehicle location equipped fleets, surveillance cameras, etc. 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 the State Departments of Transportation 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 large, multivariate datasets at reasonable speeds (Amin-Naseri et al. 2018; Richardson et al. 2014). Traditional data warehouses are stretched to the limit due to the enormous size and speed and 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. In the above context, the primary goal of this research is to deliver a prototype design and deployment of TITAN, an interactive web-based framework for storing, retrieving, integrating, analyzing, and visualizing big 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, heavily reliant on relational databases housed in big, costly enterprise machines.

### **1.3 Objectives of the Study**

The objectives of this study can be summarized as follows:

- 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.
- Offer a low-cost but effective data integration and analytics platform by leveraging open-source software for designing, developing, and deploying TITAN.

- Develop a scalable architecture for data storage and processing and leverage parallel processing to analyze data at fast speeds.
- Leverage GPUs (Graphical Processing Units) for data visualization.
- Provide user-centered, web-based data visualization to enable easy interaction with the platform and provide users with quick access to the integrated datasets via a user-friendly, interactive web interface.

## **CHAPTER 2: LITERATURE REVIEW**

Before beginning the proposed methodology section, the literature review has been conducted on the interactive web-based platform for transportation data integration and analytics. The literature review helps to understand the growth of this research area and identify the challenges or issues are faced by other researchers.

In general, data integration should be conducted before data analysis. Data integration is a complex challenge for organizations that deploy big data architecture due to the heterogeneous nature of data used by them (Kadadi et al. 2014). Despite the insurmountable growth of data in big data scenarios, users usually look for a unified view of the data available from heterogeneous data sources; therefore, integration issues are increasingly garnering attention (Abbes and Gargouri 2016). Data integration is concerned with unifying data that shares common semantics but originates from unrelated sources, referring to combining data such that a uniform view is available to users (Abbes and Gargouri 2016). While working on data integration, it is essential to deal with heterogeneity, as it creates an interoperability problem when distributed systems need to cooperate. In order to solve this problem, both structural and semantic heterogeneities must be dealt with (Malucelli and Oliveira 2003; Abbes and Gargouri 2016). According to Dong and Srivastava (2013), big data integration differs from traditional data integration in many dimensions: (i) the number of data sources, even for a single domain, has grown to tens of thousands, (ii) many data sources are highly dynamic, as a huge amount of newly collected data is continuously made available, (iii) the data sources are extremely heterogeneous in their structure, with considerable variety even for substantially similar entities, and (iv) the data sources are of widely differing qualities,



with significant differences in the coverage, accuracy, and timeliness of data. However, the authors only explored the progress made by the data integration community in schema mapping, record linkage, and data fusion and not how to deal with big data integration based on the aforementioned four dimensions.

Big data is becoming a research focus in intelligent transportation systems (ITS), as seen in many projects around the world (Zhu et al. 2018). Besides ITS, big data analytics is used in many areas such as machine learning, computer vision, and web statistics (Ayed et al. 2015). Today, many big data analytic solutions are available, but the most used is the open-source Apache Hadoop framework. Hadoop uses a distributed storage and parallel computation model over a cluster of many commodity machines to easily handle big data (Ayed et al. 2015). According to Vlahogianni (2015), traditionally, turning data into knowledge relies on classical statistical analysis and interpretation; this fundamentally requires analysts to become intimately familiar with the data and serve as an interface between the data and users. With the recent availability of very large data sets (big data), this form of manual probing becomes slow, expensive, and frequently unfeasible (Vlahogianni 2015). Therefore, the authors think new approaches are needed to efficiently deal with some of the challenging issues related to big data such as the data size, high dimensionality, and overfitting. Finding more efficient big data analytical approaches is necessary, as big data can be generated anywhere using any digital device. It can be produced by cell phones, social media, sensors, transactional systems, vehicles, industrial machines, PCs, satellites, and cameras that monitor traffic (Guido et al. 2017). In their work, the authors introduced the capillary diffusion of wireless technologies and the entire network infrastructure, which allow to detect and collect large amounts of

spatio-temporal data that can be used to understand patterns and innovative interpretative models that, in the specific field of mobility, can direct urban planning, sustainable mobility, and transport engineering. To improve decision-making capacity and problem-solving ability reliably and in real time, their study aimed to present a Decision Support System (DSS) framework aimed at proposing travel strategies alternative to individual modes by elaborating a large amount of transportation systems data coming from different devices. The proposed DSS framework tends to fill the gap and create preconditions necessary to improve the modal split in favor of public transport, which has not yet implemented all its functions by the Centrale Operativa Regionale (C.O.RE.). Their research focuses on how to build the public transport system, but it is not very user-friendly. The authors did not discuss and provide more details about big data visualization that is necessary to help users better understand the data and results.

Big data visualization has proven to be effective for not only presenting essential information as vast amounts of data but also driving complex analyses (Keim et al. 2013). Big data analytics and discovery present new research opportunities to the computer graphics and visualization community (Keim et al. 2013). Effective data visualization is the bridge between the quantitative content of the data and human intuition and thus an essential component of the scientific path from data into knowledge and understanding (Donalek et al. 2014). Moreover, visualization is essential in the data mining process, directing the choice of the applicable algorithms, and in helping to identify and remove bad data from the analysis (Donalek et al. 2014). In their work, the authors try to explore the use of immersive virtual reality platforms for scientific data visualization, both as software and inexpensive commodity hardware. Although their platform is not used for

the transportation of big data visualization specifically, exploring techniques and methodology such as the application, some software libraries, and building tools in their research are still worthy to learn. According to Ali et al. (2016), big data visualization has become the topic of interest for all industries but faces quite a few challenges; for instance, big data visualization tool must be able to deal with semi-structured and unstructured data because big data usually has this type of format. Their work also proposed several visualization tools such as Tableau, Microsoft Power BI, Plotly, Gephi, etc.

Several studies are related to big data visualization, integration, and analytics but not many on building an interactive web-based platform for combining all the transportation big data integration, visualization, and analytics works. Researchers are still working on it. During the past decade, various web-based archived data user service systems/platforms have been developed in attempts to increase the exchangeability and usability of data (Ma et al. 2011). For instance, Li et al. (2018) combined the existing LiDAR processing tools with Hadoop to handle the high computational intensity of LiDAR data. Their framework can conduct data processing in parallel with high-scalable distributed computing speed. Prasad and Agarwal (2014), on the other hand, introduced a framework called SAMOA (Scalable Advanced Massive Online Analysis), which is an open-source framework based on JAVA and supports several distributed processing platforms plugging into it. SAMOA allows users to use the existing machine learning directly or developers can develop their new algorithms rapidly. As mentioned above, the three challenges of big data while conducting stream mining are volume, velocity, and variety. The authors supposed that most solutions existing today solve at most two

challenges. Their platform SAMOA is a recent and open-source framework for distributed machine learning that addresses all three challenges while conducting big data stream mining (Prasad and Agarwal 2014). Another framework worth mentioning is MapReduce, a new parallel processing framework based on open-source Hadoop. Compared with the existing parallel processing data analysis tools, according to Mohammed et al. (2014), it has two advantages: 1. fault-tolerant storage resulting in reliable data processing by replicating computing tasks and cloning data chunks on different computing nodes across the computing cluster and 2. high-throughput data processing via a batch processing framework and Hadoop distributed file system (HDFS). Data is stored in HDFS and made available to slave nodes for computation. Like other researchers, this project aims to create an interactive and multi-functional system but aims to focus more on developing a framework for storing, retrieving, integrating, analyzing, and visualizing big transportation datasets.

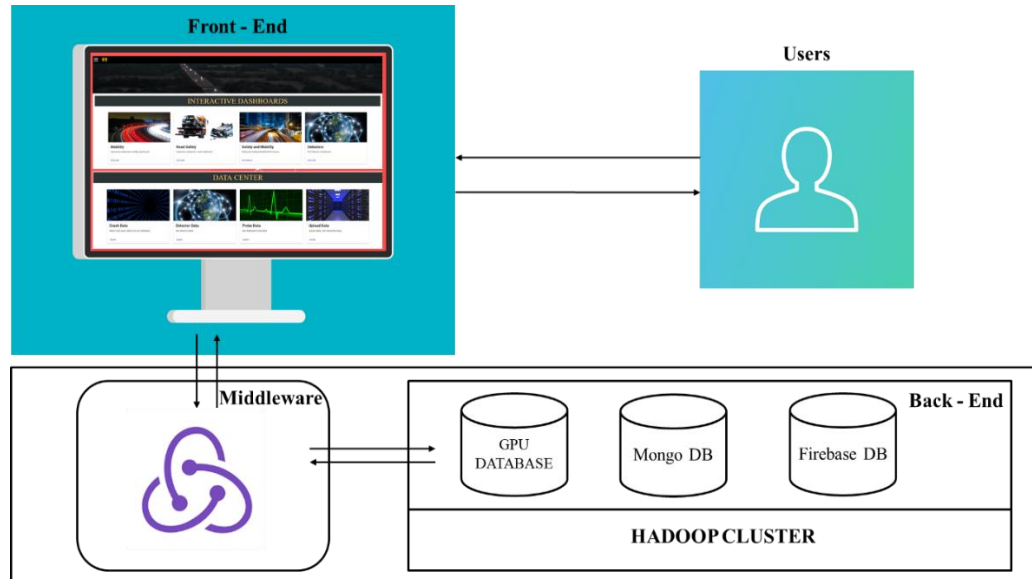
The content of the paper is mainly divided into three chapters: Chapter 3 discusses the proposed methodologies. This chapter discusses the design of the TITAN framework, the main structure of the TITAN platform, and the five main TITAN applications. Chapter 4 discusses the performance evaluation of the platform, comparing TITAN with other data processing platforms such as Oracle and Tableau. Chapter 5 discusses the conclusions of this project and the prospects for future research. Finally, Chapter 6 demonstrates some sample applications of TITAN.

## CHAPTER 3: PROPOSED METHODOLOGY

The methodology is developed to analyze the TITAN's design approaches and main applications. TITAN has four main components, which have been discussed in detail in the subsequent paragraphs: Back End, Front End, Data Center, and APPCENTER.

### 3.1 TITAN Design Approach

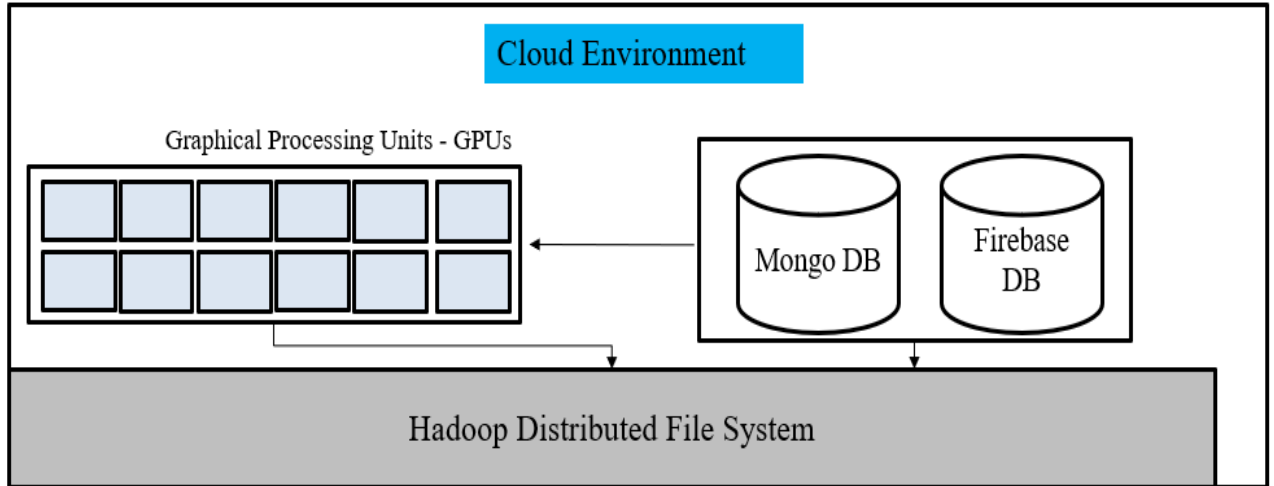
TITAN's design architecture seeks to address key technological gaps in data handling, archiving, and analysis for decision support. The following section provides a detailed overview of its components. Figure 1 illustrates the design framework that enables TITAN to handle fast, disparate, noisy data streams in a seamless manner. It comprises 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.



**Figure 1. Schematic of TITAN's Key Components**

### 3.1.1 Back-End 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 five years. It is expected that the back end can 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. At the core of the back end is an HDFS (Shvachko et al. 2010), which enables the networking of a series of computers into clusters. Using the HDFS enables maintaining the processing speed of TITAN even as the size of the data grows. On top of the Hadoop framework are three different databases: Firebase (Tanna and Singh 2018), MongoDB (Chodorow 2013), and a GPU database. Their roles are defined as follows.



**Figure 2. Back-End Design Framework**

### 3.1.1.1 Firebase DB

Firebase is a NOSQL database for storing datasets that are not structured. The NOSQL means non-relational in contrast to traditional Structured Query Language (SQL) databases. Moreover, NOSQL 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 an 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. Further, Firebase is used to temporarily store all user-uploaded datasets. As Firebase is a NOSQL database, it can consume all types of data formats: Shapefiles, CSVs, XML, Video, Images, etc. In contrast, an 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 HDFS and MongoDB depending on the use of the data.

### 3.1.1.2 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 3 provides an example of how data is stored in Mongo. Each row of data is stored uniquely with an identifier (\_id). As data is stored by rows and not as one big table, Mongo can store files with an 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 3. Data Storage in Mongo DB

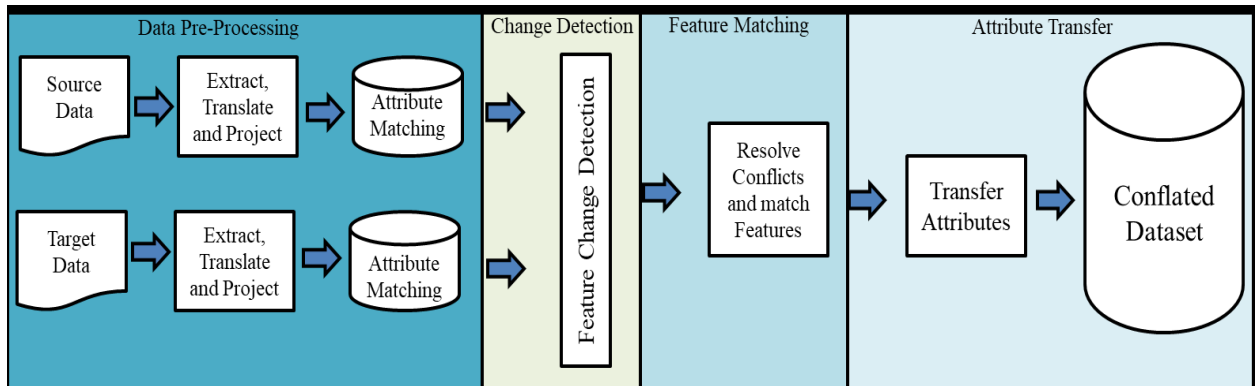


### **3.1.1.3 Graphical Processing Unit (GPU) DB**

All front-end visualizations are carried out by using an open-source GPU database. GPUs have tremendous processing capabilities compared to CPUs. For example, a single NVIDIA GeForce 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 TITAN is highly fast on the front end. A SQL database is used on top of this database to process data in the memory of the GPU.

### **3.1.2 Data Integration Layer**

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. 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. Automated Spatial Data Conflation Process**

### 3.1.2.1 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.

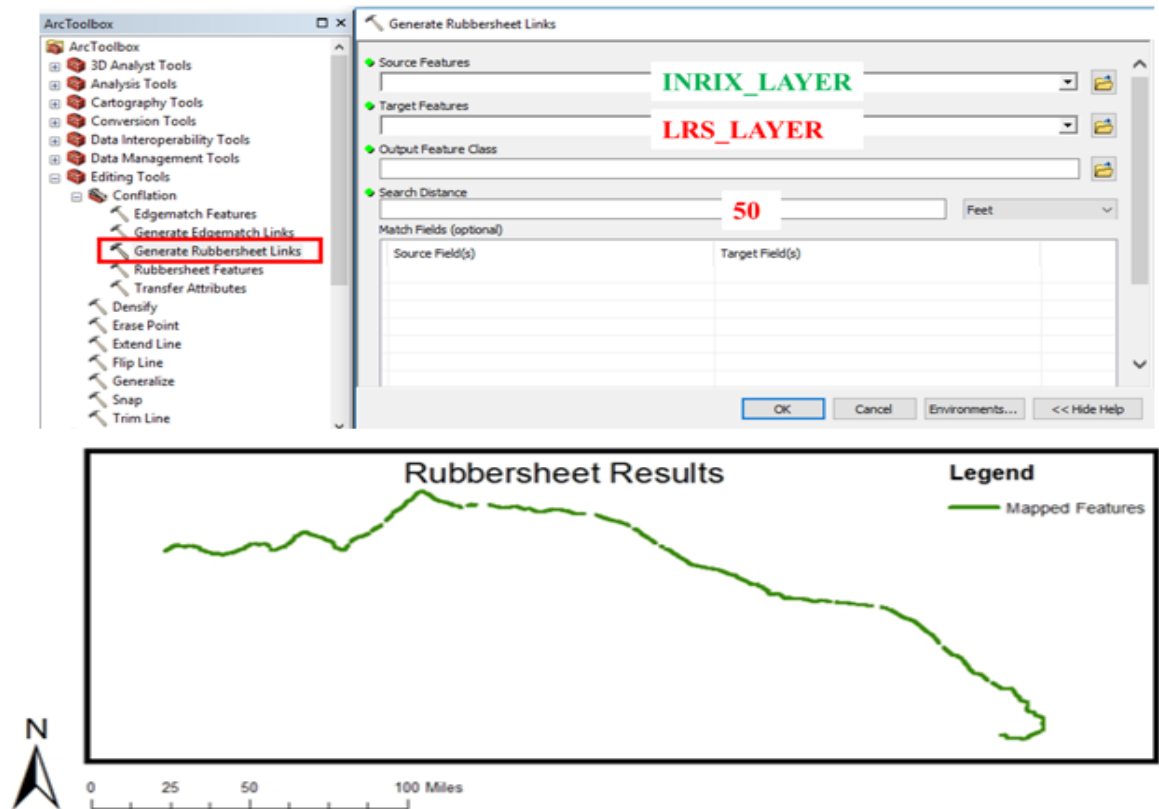
### 3.1.2.2 Detect Feature Changes

Feature change detection is the second step of the conflation process. Knowing where and what the changes are between the two datasets helps to assess how significant they are and whether to proceed with attribute transfer. To detect feature changes in two datasets, the Detect Feature Change (DFC) tool in ESRI's ArcGIS was used. This tool identifies spatial feature differences and outputs the type of change detected for each feature. Ideally, four possible conditions 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.

### **3.1.2.3 Feature Matching**

The goal of feature matching is to map features in source datasets, which may have experienced a spatial change to its corresponding target features (base layer). In this study, we mainly used two types of data: MO-DOT Linear Referencing System (LRS) data and INRIX road data. The ESRI's ArcGIS feature matching tools were used in this project. They 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 5. It generates links between matched features or points where the source and target locations are identical. In Figure 5, the gaps in the rubber sheet results frame (green layer) shown represents regions where road segments were not matched. The conflation rate does fluctuate depending on the type, geometry, and length of the road segment.



**Figure 5. ArcGIS Rubbersheet Tool and Results of Conflation. Gaps Indicate Road Sections Missed**

#### **3.1.2.4 Transfer Attributes**

Finally, once the features between the two geographic data layers have been matched, specific feature attributes from the source layer are transferred to the matching target features. Table 1 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 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

### 3.1.3 Conflation

Conflation found its first application around the mid-1930s, but it was not popular among researchers until the late of 1980s. Conflation is often a term used to describe the integration or alignment of different geospatial datasets (Chen et al. 2006). Conflation is the process of combining the information from two (or more) geodata sets to make a master data set that is superior to either source data set in either spatial or attribute aspect (Yuan and Tao 1999). Initially, the primary goal of conflation was to eliminate any sort of spatial inconsistency from the various vector maps to achieve a desirable accuracy.

For this study, data available from free sources as well as State DOT's data were used for the analysis. There were primary four types of data: INRIX (probe data), crash data, transit data, and detectors data. They contain different information. Conflating each of them can lead to more and accurate results.

### 3.1.3.1 Link Four Types of Data

**Conflate probe data and crash data:** Probe data provides information on travel time and speed, and crash data can tell us about congestion. Linking both gives two results. The first is whether the crash creates congestion. If there is only crash data without travel speed and time, we cannot judge whether there is congestion. Secondly, it can help us to tell whether congestion causes the rear-end crash. For example, if there is a queue about which the driver does not know, then he would come and hit the back because of the sudden speed differentials. Therefore, congestion happens before the crash, meaning that the congestion causes the rear-end crash.

**Conflate crash and detector data:** Detector data provides information about the volume, occupancy, and speed at a point. When volume and speed are linked with the crash, it can inform how many people are affected by the crash. When the crash occurs, the number of people who passed by after the crash are those affected by it.

**Conflate detector and probe data:** Probe data provides information on travel time and speed but not volume. If only probe data is available, one cannot tell whether there is congestion. For instance, if we only know that the travel speed is 20 miles/hour but don't know the number of vehicles, we cannot tell whether there is congestion. If the speed is 20 miles/hour, it could be of just one car driving on the road that is driving very slowly, which does not mean there is congestion. However, on linking the volume information the volume reveals that there are 200 cars on the road and all are driving slowly; we can then conclude that there is congestion. Tying in volume and speed can lead to knowing whether congestion occurs.

**Conflate crash and transit data:** Transit data can give us information about the bus

route and travel time. Linking both data helps to know whether there is a crash and how they can route their buses. If a crash occurs, we know what crash occurs and which buses are affected; we can then rearrange the bus routes. Additionally, it can help us to know the risk of a bus route. If too many accidents happen on a route, then it means the risk on this route is high.

After all conflations, they are tied into the LRS system. The LRS is the map of all the routes in the State, so we need to know how the probe and crash connect to LRS. Most agencies such as DOT and people use LRS shapefile rather than INRIX. What they want to know is where the crash happens on the LRS rather than where the crash happens on your probe segments. This is why we conflate them to the LRS system. The LRS is like a base map and everything in on top of it.

### **3.1.3.2 Three Types of Conflation**

Conflation can be divided into three types. The first is point to line, the second is line to line, and the last is milepost to line.

#### **3.1.3.2.1 Conflate Point to Line**

Conflation point to line was used for transit application. Points represent the bus stops and lines represent the road segments from probe data. Generate Near Table, which is a tool of ESRI's ArcGIS Proximity toolset, was used for conflation. INRIX data is the input feature, and LRS data is the nearest feature. We used it to find three nearest segments to the stop within 100 feet and then conflate the nearest line to the stop. The output table contains the proximity and other attributes.

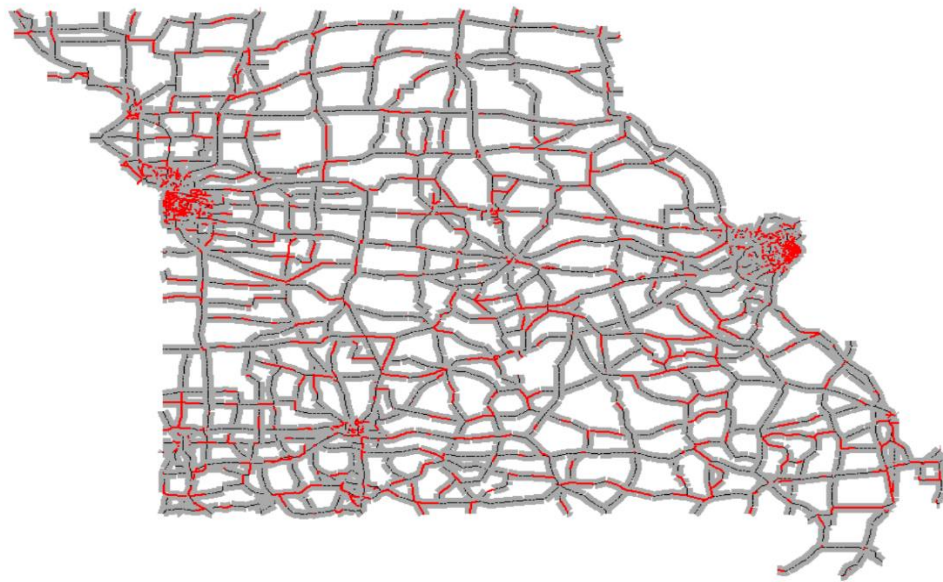
#### **3.1.3.2.2 Conflate Line to Line**

For safety and mobility application in TITAN, we conflated data line to line. The

first step of conflation is road conflation. At first, we only used the SequenceMatcher function, with only 350 pairs of output. Following this, the similarity function was added to increase the output of the matched pairs. The similarity function has been provided as follows:  $\text{Value} = () * \text{sequence} + () * \text{numberCheck} + () * \text{exDirection}$ . The front coefficient number can be adjusted to achieve the goal wherein more correct pairs and fewer wrong pairs can be the output. The final one used in this research was  $\text{Value} = 0.1 * \text{sequence} + 0.75 * \text{numberCheck} + 0.15 * \text{exDirection}$ , which output 1296 pairs road.

The next step after road conflation is segment conflation. Generate Near Table was used in this step to find the number of near features reported for each input feature within a limited radius. In this research, for segment-to-segment conflation, we found three nearest segments in 100 feet. For segments to stops, we found 1000 nearest stops in 100 feet. Both INRIX and LRS provide information on the quadrant, which is used to check the direction. Moreover, a bearing check can be used to check the road direction. The final step of conflation is to input both the paired and unpaired segments into the ArcGIS to get the ArcMap. The ArcMap is shown in Figure 6. The grey layer shows the paired segments and the red shows the unpaired segments.





**Figure 6. ArcMap of Paired and Unpaired segments**

#### 3.1.3.2.3 Conflate Milepost to Line

Milepost-to-line conflation is used for TITAN crash data. The detector data has no information on the coordinates of points. Therefore, to know the location of the points, we need to conflate milepost to line. First, as segments of probe data provide information on the coordinates, join all probe segments together as one line in one layer in the ArcGIS. Divide the line into miles with mile markers on it. Then convert everything to miles and transfer the coordinates to one whole route that comprises only miles. The next step is to locate each segment, i.e., its start and end latitudes, and then locate where the mile markers close. A point on the line can be found between the start and end latitudes of one segment, and then, the point and the segment can be mapped together. For example, there is a point on mile marker 15 between the start and end latitudes of segment 123. Then mile marker 15 would be mapped to segment 123.

In a peculiar circumstance, when a point is exactly on the breakpoint of two

segments, it is mapped twice. On conflation, if it has two segments, both segments are used.

### 3.1.4 Front-End Design

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 keep changing constantly. Thus, 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 lists the most popular front-end development frameworks, the number of contributors, and their 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. Choosing a Front-End Library for TITAN's Development**

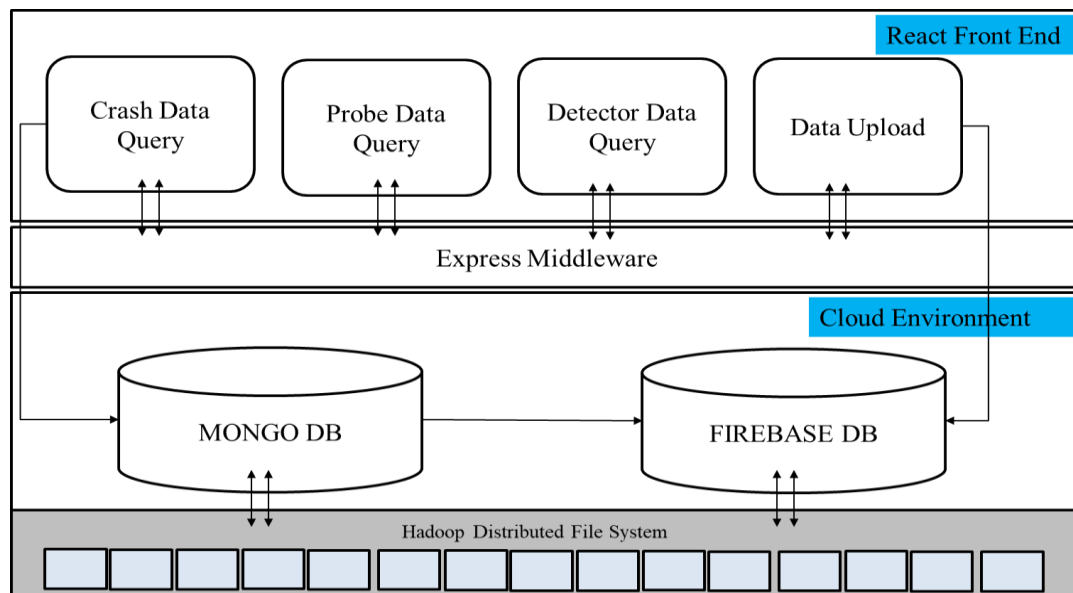
Front-End Library	Number of Contributors	GitHub Popularity
React	1285	124,747
Aurelia	97	10,873
Angular	1596	59,434
Ember	753	20,772
Vue	268	131,290

TITAN was developed with React (Abel 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 the number 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.

### 3.1.5 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 7. The main resources used include a Hadoop cluster, which stores all 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 being pushed to the cluster.



**Figure 7. 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. Moreover, it hosts a single application for uploading different types of data into TITAN's databases. Figure 8 shows all the applications deployed in the Data Center.



**Figure 8. An Interactive Dashboard of Data Center**

#### **3.1.5.1 Data Upload**

Sharing data on TITAN is a straightforward process. A layout of the form for uploading data into TITAN is shown in Figure 9. The information requested helps to build applications driven by user or agency input. Users are required to provide the name of the agency sharing the data, its type of data, possible uses, and limitations. Once the 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.

Enter your name

Enter the title

Select Data Type

SELECT YOUR FILE

**B** *I* U ~~ABC~~ {} x² x₂ Normal ▼ 16 ▼ Font ▼

☒ 🔍 ↺ ↻

Add a tag

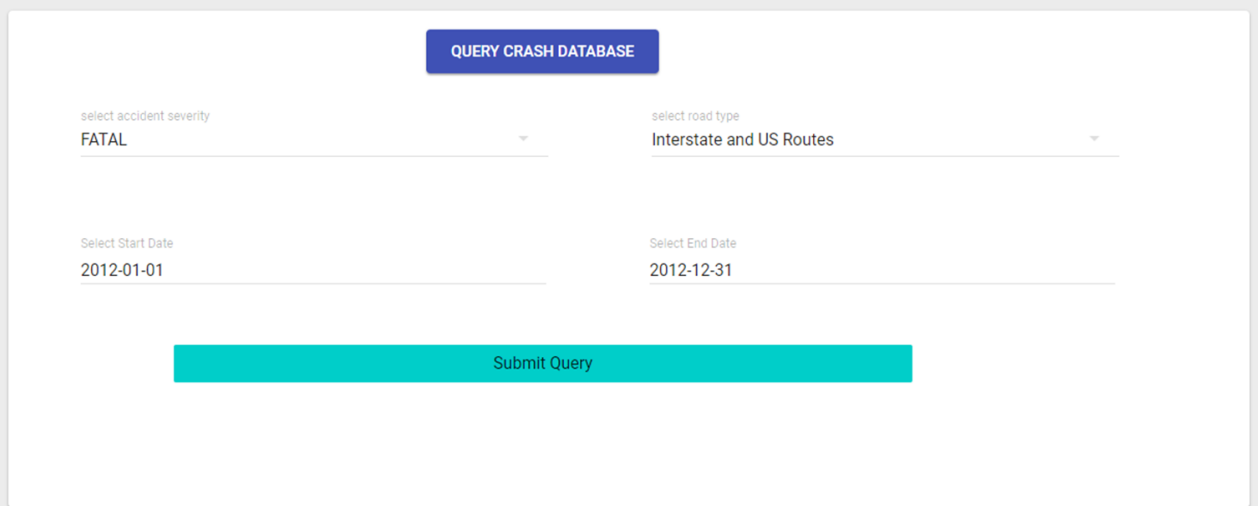
UPLOAD DATA

**Figure 9. Uploading Data to TITAN**

### 3.1.5.2 Data Query

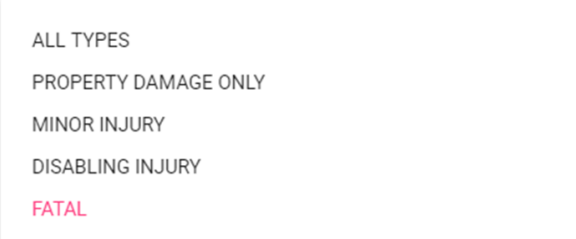
Moreover, the Data Center permits users to query data from different sources such as crash, detector, and probe data. The non-programmatic interface is enabled with functions that help users select and filter their respective areas of interest. The output of each query is a downloadable comma-delimited (CSV) file containing all the information requested by the user. The simplicity of the CSV file allows data to be inputted into a wide range of software such as spreadsheets and database clients. Figure 10 provides an example of an 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 11 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 12–15.



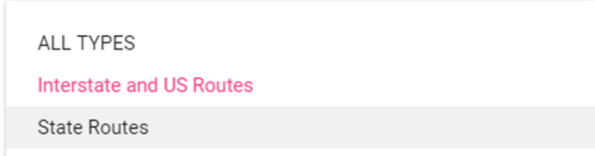
The screenshot shows a web interface for querying a crash database. At the top center is a blue button labeled "QUERY CRASH DATABASE". Below it are four input fields arranged in a 2x2 grid. The top-left field is labeled "select accident severity" and has "FATAL" selected. The top-right field is labeled "select road type" and has "Interstate and US Routes" selected. The bottom-left field is labeled "Select Start Date" and has "2012-01-01" entered. The bottom-right field is labeled "Select End Date" and has "2012-12-31" entered. At the bottom center is a large teal button labeled "Submit Query".

a).



A vertical list of accident severity options. The options are: "ALL TYPES", "PROPERTY DAMAGE ONLY", "MINOR INJURY", "DISABLING INJURY", and "FATAL". The "FATAL" option is highlighted in pink.

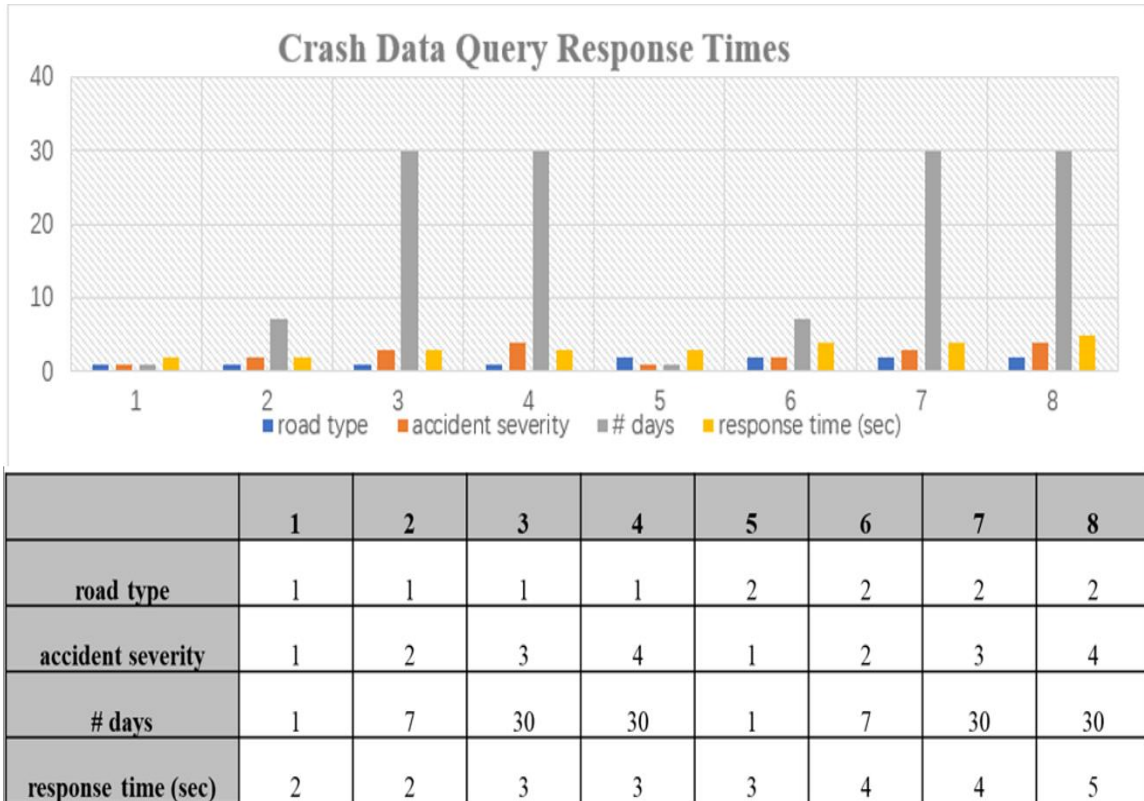
b).



A vertical list of road type options. The options are: "ALL TYPES", "Interstate and US Routes", and "State Routes". The "Interstate and US Routes" option is highlighted in pink.

c).

**Figure 10. Crash Data Query Interface**



**Figure 11. 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)**

QUERY DETECTOR DATABASE

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

Submit Query

QUERY DETECTOR DATABASE

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

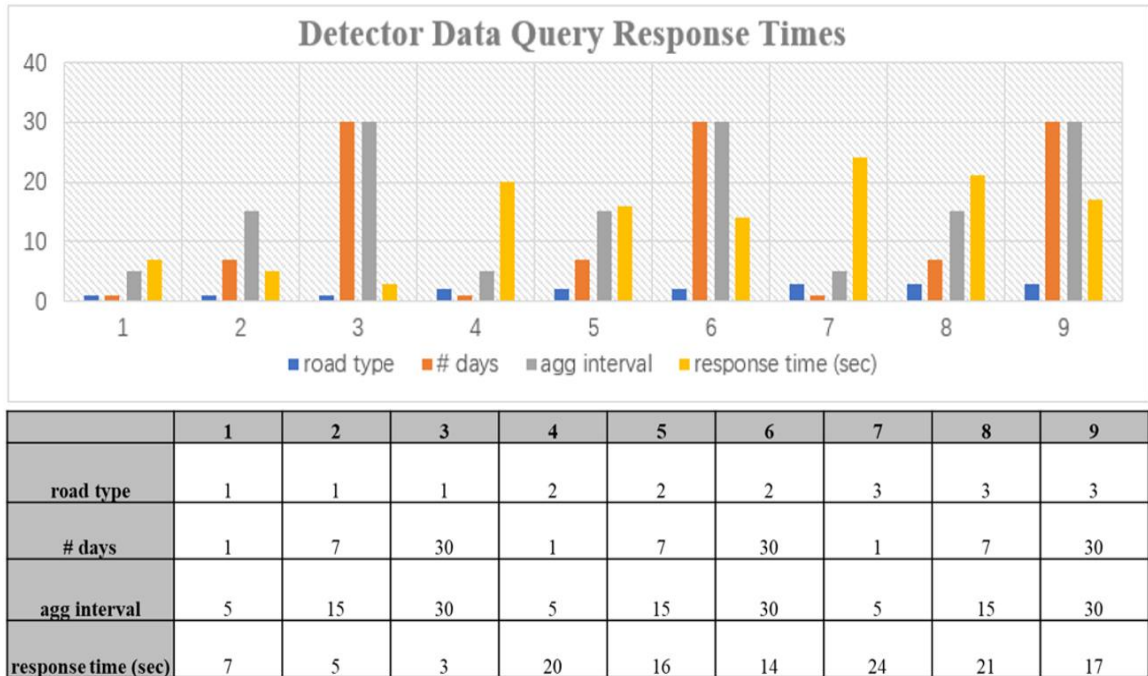
☒ 30-minutes

Submit Query

Download CSV

**Figure 12. Detector Data Query Interface**





**Figure 13. Detector Database Query Times: Road Types – Freeways (1), Interstates (2), All Road Types (3)**

QUERY PROBE DATABASE

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

Submit Query

QUERY PROBE DATABASE

select county  
CLAY

select direction  
NORTHBOUND

Aggregation Interval  
☒ 5-Minutes  
☐ 15-Minutes  
☐ 30-Minutes

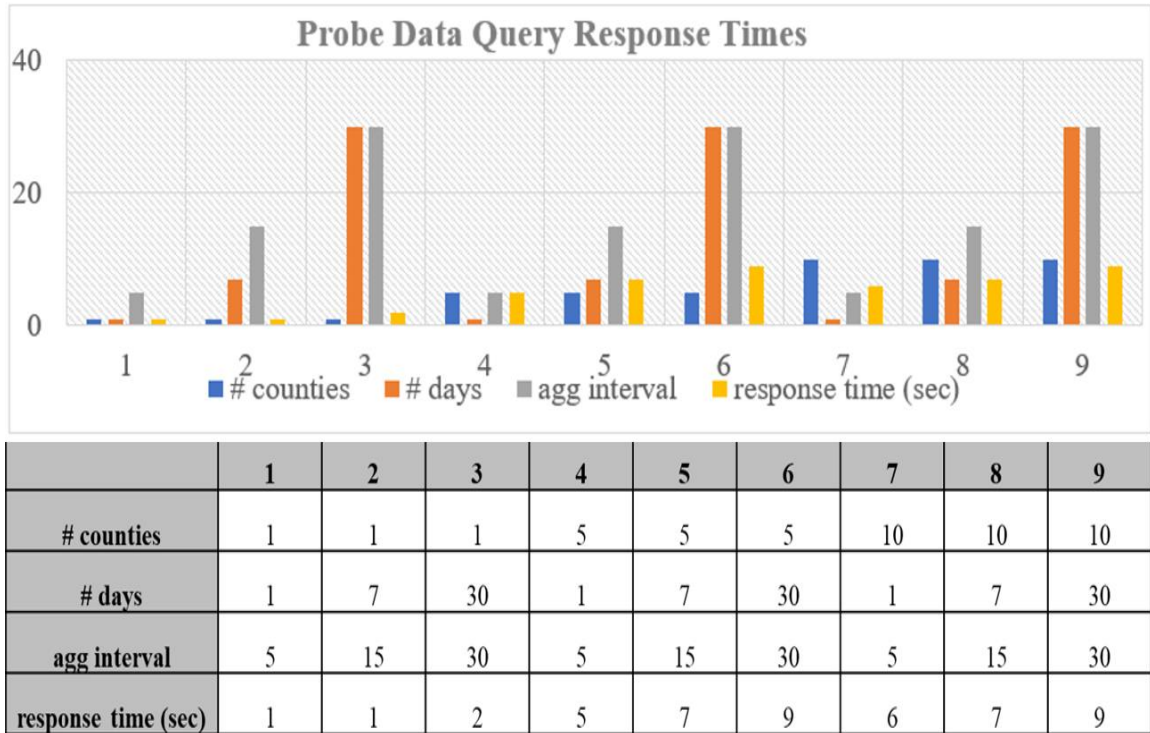
Select Start Date  
2018-01-01

Select End Date  
2018-01-02

Submit Query

Download CSV

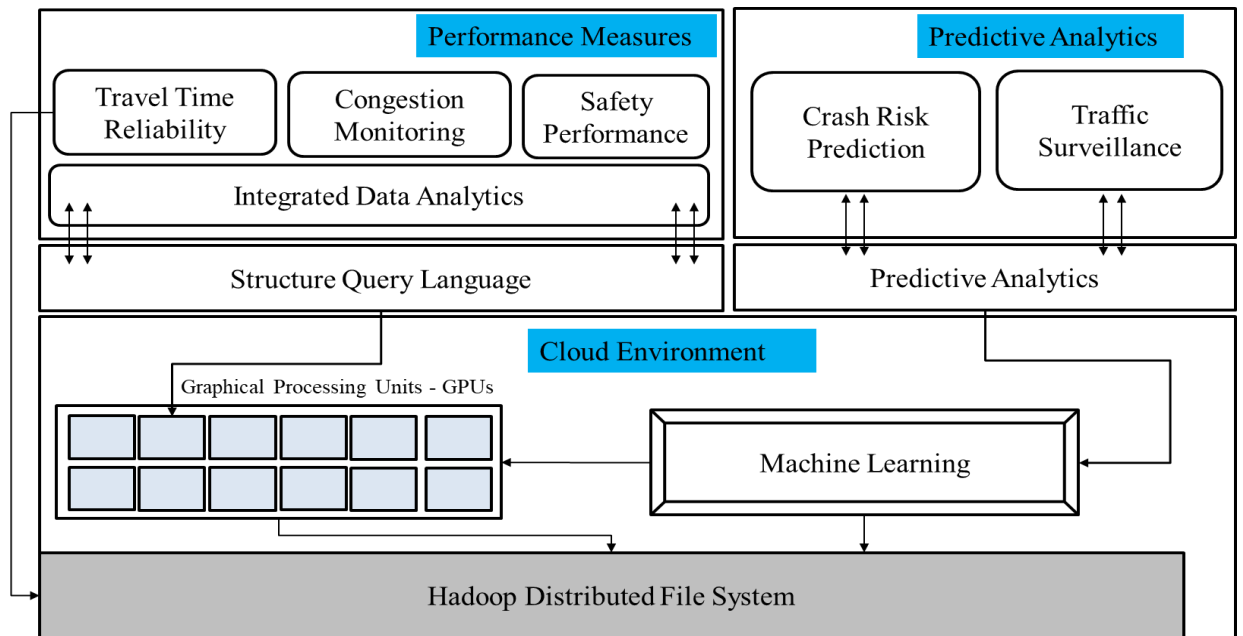
**Figure 14. Probe Data Query Interface**



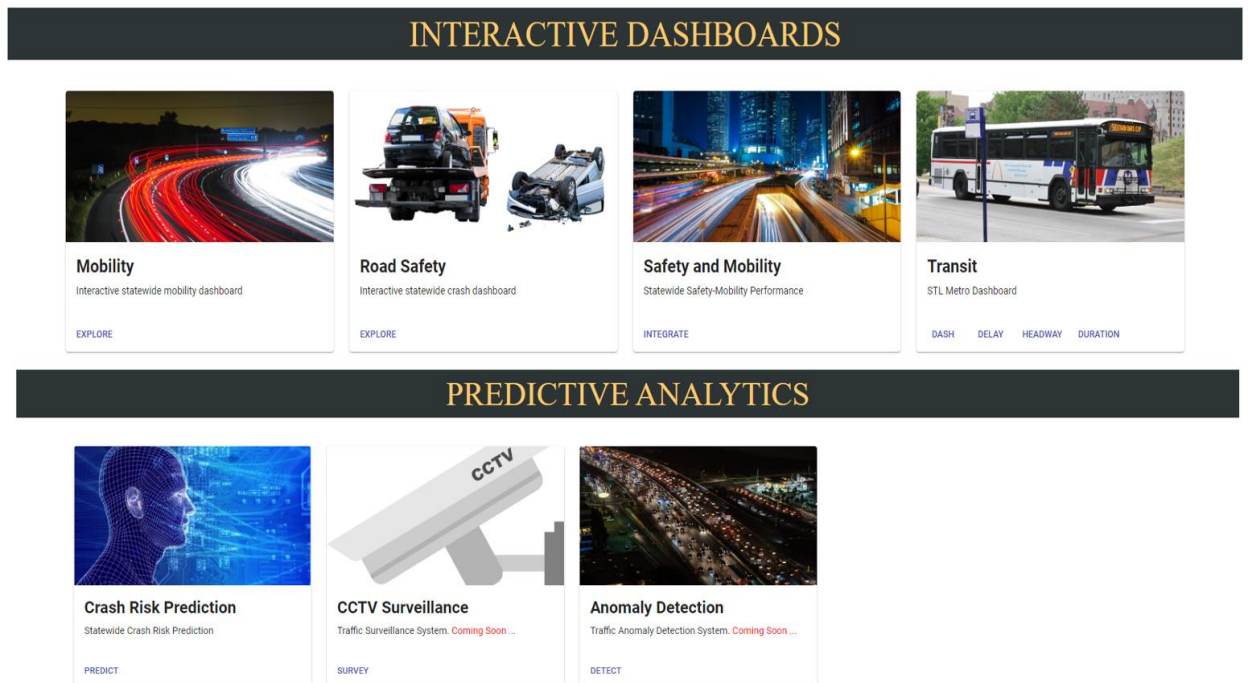
**Figure 15. Probe Database Query Times: Aggregation Interval in Minutes**

### **3.2 TITAN Application**

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 the APPCENTER, as shown in Figure 16, 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 and 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 17 shows different applications that have been developed in the APPCENTER. In this research, four application examples are shown: Mobility, Transit, Safety and Mobility, and Predictive Analytics.



**Figure 16. APPCENTER Design Framework**



**Figure 17. Applications Center – APPCENTER**

### 3.2.1 Mobility

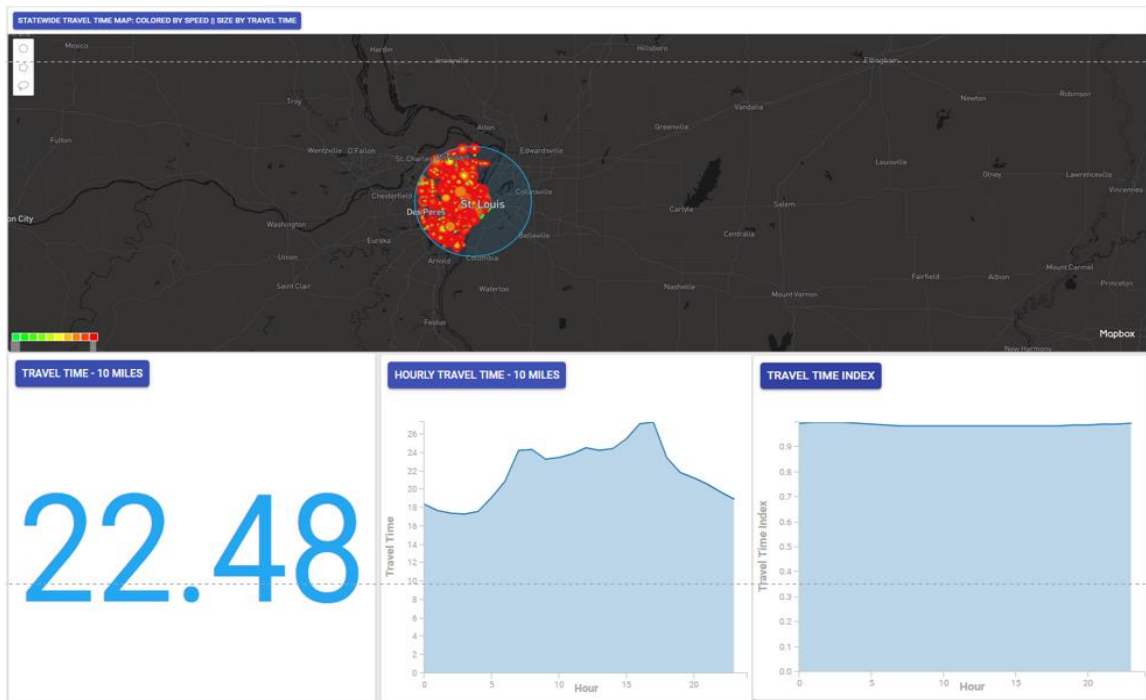
Big data is often defined in terms of large Vs: volume, variety, and velocity. Table 3 shows the volume, variety, and velocity of datasets currently stored in TITAN's Data Center. Statewide probe data, which constitutes the largest share, had about 80GB generated annually. TITAN currently has two years of probe data stored. At an aggregation interval of five 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. Volume, Velocity, and Variety of Datasets Archived on the TITAN Platform**

<b>Dataset</b>	<b>Size</b>	<b>Rate</b>	<b>Period</b>
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		

Although the data is very large, the TITAN platform can handle this big data quickly. Mobility 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 18 GB. APPCENTER can visualize and perform queries on this data without latencies in the web browser. Users can use this dashboard to get travel time and congestion hours.

In addition, TITAN dashboards support user-friendly interactive operations for users. Users can zoom in, zoom out or use a circular and lasso filter to select regions of interest from the map chart. And the accompanying charts could be used to filter and select various areas of interest. Figure 18 shows the statewide travel time map chart and several accompanying charts in interactive mobility dashboard. The map chart shows travel time of segments colored by speed and the size of the dots represents the segment travel time. In this figure, the area near to St. louis was selected, then, the accompanying charts refreshed to show relevant information in this area. The travel time chart shows the average travel time of this segment is 22.48 minutes per 10 miles of this area. The hourly travel time chart indicates how travel time changes as time pass. The travel time index chart displays the ratio of the travel time during the peak period to the time required to make the same trip at free-flow speeds which provides users the peak hour information.



**Figure 18. An Interactive Dashboard for Exploring Statewide Mobility Trends**

### 3.2.2 Transit

Transit is another important application of TITAN. The transit dashboard is designed for assessing the performance of transit systems such as bus lines or evaluating accessibility issues related to transit. The transit application required conflation of both transit and mobility data. That enables the system to compute the reliability of bus routes based on traffic conditions. The conflated data had around 98 million rows. Figure 19 shows three charts from transit visualization dashboard. From the figure, it can be seen that in the bottom left average duration chart, the top three bus lines were chosen. Other accompanying charts and the map chart were refreshed for the top three bus lines. The bus location map in this interactive dashboard shows the trajectories of these 3 bus lines colored by trip duration and sized by delay. Another average delay chart indicates that the average delay of these three bus lines has a sudden increase in peak hours.





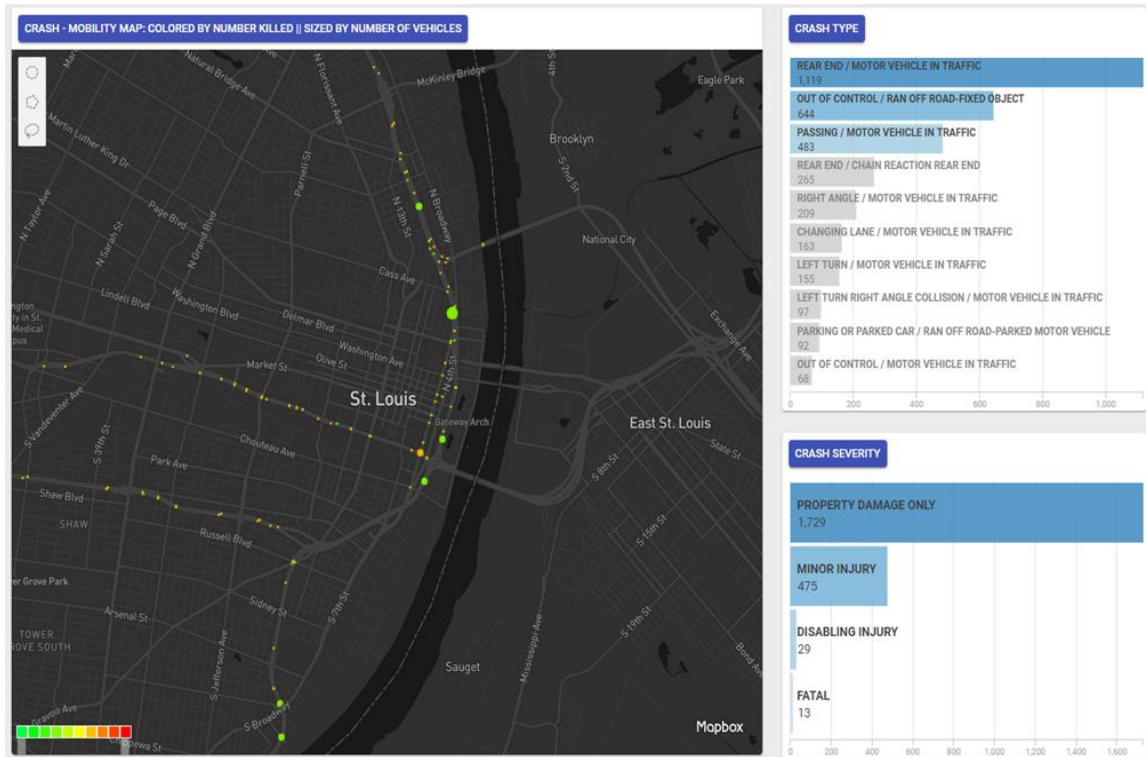
**Figure 19. An Interactive Dashboard for Exploring Transit Dashboard**

### 3.2.3 Safety and Mobility

The impact of road crashes on mobility or vice versa is very important for estimating the cost of a crash or the benefits of mobility improvements. The safety and mobility application required conflation of both crash data and mobility data and the result is a mapping between probe segments and accident locations. The conflation of both datasets resulted in a unified data with 246 million rows which were consumed by the framework for visualization.

The interactive operations are similar with the other two application dashboards. For example, there are three charts from the safety and mobility dashboard shown in figure 20. The chart at the top right represents the number of crashes of each crash type.

The top three crash types were filtered. The map chart displays the locations of crashes associated with road segments for these top three crash types. The size of the dots represents the number of vehicles and the color represents the number of killed. And another chart represents the number of crashes of each crash severity of these three crash types.



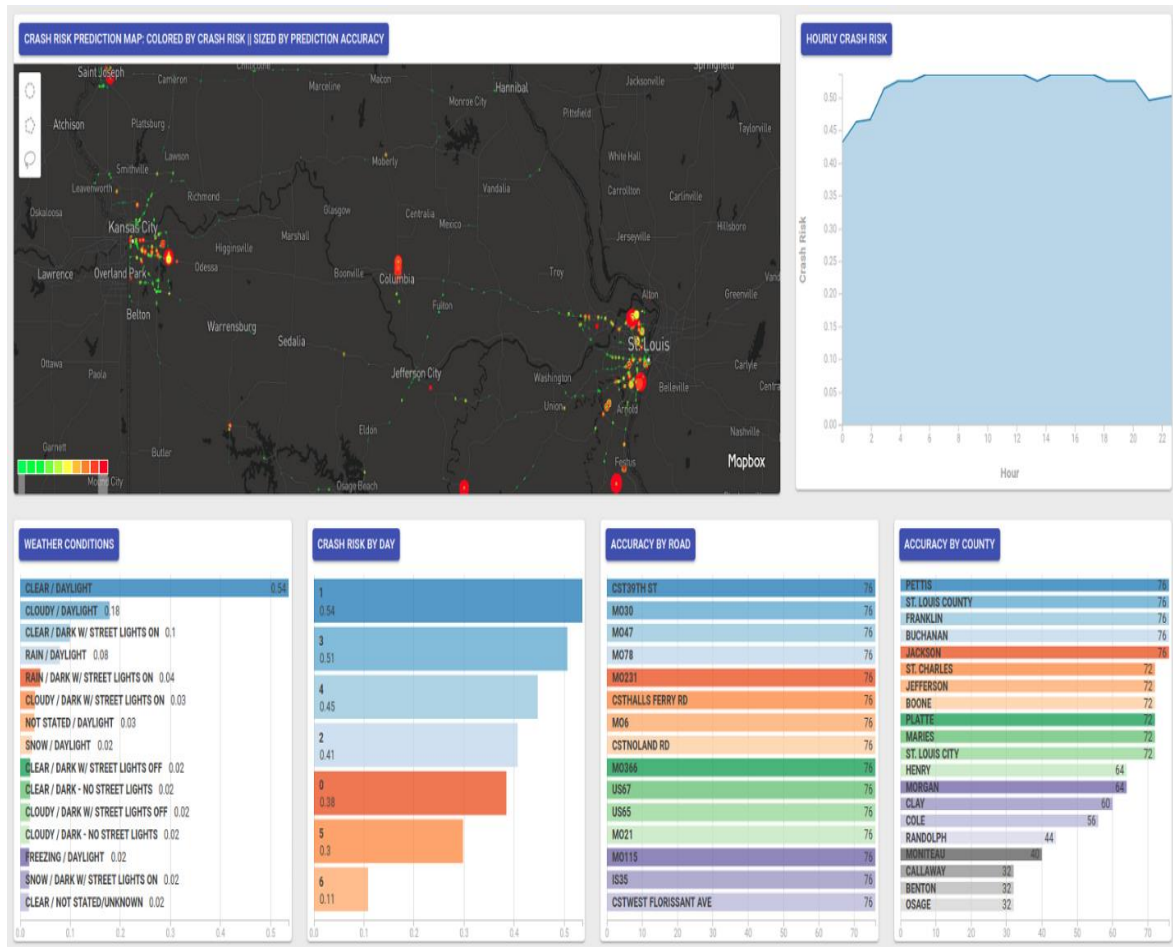
**Figure 20. Statewide Safety-Mobility Dashboard**

### 3.2.4 Prediction Analysis

Moreover, TITAN has the capability to learn from historical datasets and predict 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.

#### **3.2.4.1 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). However, 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 (daylight, night, cloudy), and historical crash trends. The current model was trained with statewide crash, probe, and weather data from 2009 to 2011 and tested on data from 2012. Visualization of predicted crashes and related accuracies is shown in Figure 21. 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 reduce accordingly.

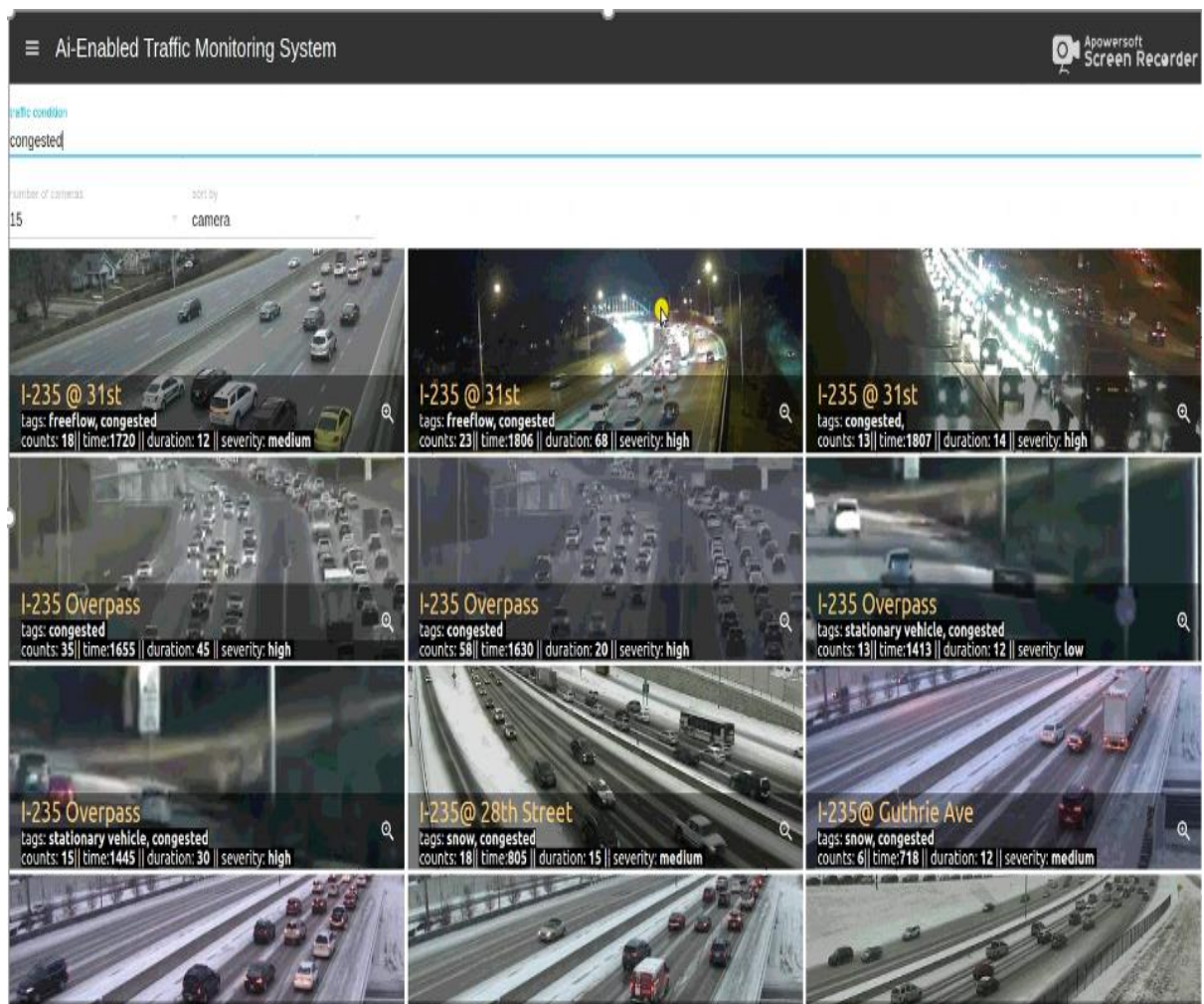


**Figure 21. Daily Predictions More Accurate than Hourly**

### 3.2.4.2 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, which is trained to detect and track various traffic incidents. This approach using computer

vision differs 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 22 shows a graphical user interface, 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 the incident.



**Figure 22. Traffic Surveillance System: Searching for Camera with Congested Scenes, Results Sorted by Camera Name**



## CHAPTER 4: PERFORMANCE EVALUATION

### 4.1 Compare with Oracle

TITAN brings together multiple databases to enable a 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. The authors compare both platforms based on expected costs and capabilities, as shown in Table 4.

**Table 4. Comparing TITAN with Traditional Data Warehouses**

	TITAN	Oracle
<b>Costs</b>		
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)
<b>Capabilities</b>		
Interactive Visualizations	✓	✓
Data Integration	✓	→
Platform Speed	↑	↓
Geospatial	✓	×
Flexibility	↑	↓
Predictive Analytics	✓	×

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 has to be expended in the development of the platform. Developers are required to spend more time on integrating all the bits and pieces of the code. This increases the development cost of TITAN compared to Oracle-based applications. Cloud costs for TITAN are expected to be slightly higher than those for Oracle because of the use of GPUs and computer clusters. TITAN will require at least two GPUs (that cost about \$0.5 an hour) and five servers in a cluster (costing about \$0.35 an hour). Oracle does not need GPUs and clusters and is thus cheaper in terms of cloud costs. Administration costs mostly depend on where the platform is deployed. On the cloud, server administration is usually done by the cloud service provider. Hence, both platforms save money. If the server is built in-house, a relational database service such as Oracle costs significantly higher. Most of the processes and applications deployed on TITAN are automated. What needs to change is the data; once the data is updated, all the apps are 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, increasing the responsibilities of the administrator, which is a relatively high cost.

Regarding capabilities, TITAN offers much more. Although interactive visualizations can be carried out on Oracle, the size of data that can be visualized is limited (not more than 2 GB). TITAN is able to visualize up to about 80 GB of data interactively without any significant latencies on the front end. Moreover, TITAN has modules for automatically integrating data from multiple sources. Data integration on platforms such as Oracle is heavily manually driven and depends 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. Furthermore, TITAN 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, all available on TITAN. Oracle-based platforms are not designed for predictive analytics.

#### 4.2 Compare with Tableau

Tableau is another interactive data visualization platform, as well as MapD. While Tableau is a CPU data frame, MapD is a GPU data frame like TITAN. Both platforms are used to process the same SQL dataset and then compare their performance, as shown in Table 5. Figure 23 shows the data visualization results by Tableau.

**Table 5. Comparison of MapD and Tableau**

	<b>MAPD</b>	<b>TABLEAU</b>
<b>Speed</b>	High	Low
<b>Convenience</b>	Easy	Complicated
<b>Data size limit</b>	40 million rows	1 million rows
<b>Cluster Compute</b>	Yes	No





**Figure 23. Data Visualization by Tableau**

In this section, Tableau is compared with TITAN in four aspects, as shown in Table 6. As both TITAN and MapD use the GPU data frame, the comparison results are similar.

**Table 6. Comparing TITAN with Tableau**

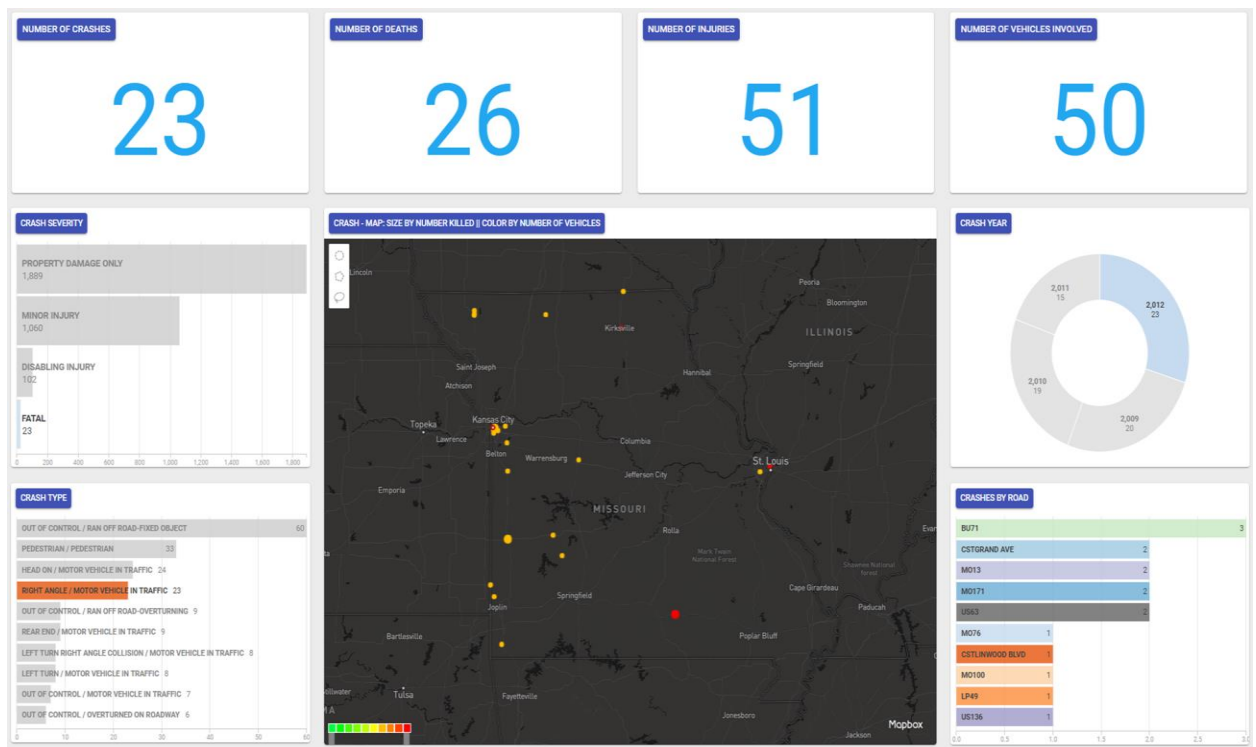
	TITAN	Tableau
Time Costs	Short time	Long time
Predictive Analysis	Yes	No
Software Costs	Low	High
Programming Need for users	No	No

TITAN is built based on several programming languages such as Java, React, etc. But for users, users can use both of these two platforms in non-programmatic environment. TITAN, as a CPU–GPU framework, uses scalable architecture for data storage and processing, leverage parallel processing for analyzing data, and leverage GPUs for data visualization and therefore requires less time for data management and

analysis than Tableau. Additionally, as mentioned above, TITAN was developed with open-source software tools, so it is cheaper than developing Tableau. Tableau is not designed for predictive analysis, but TITAN has the function of that.

## CHAPTER 5: SAMPLE APPLICATIONS

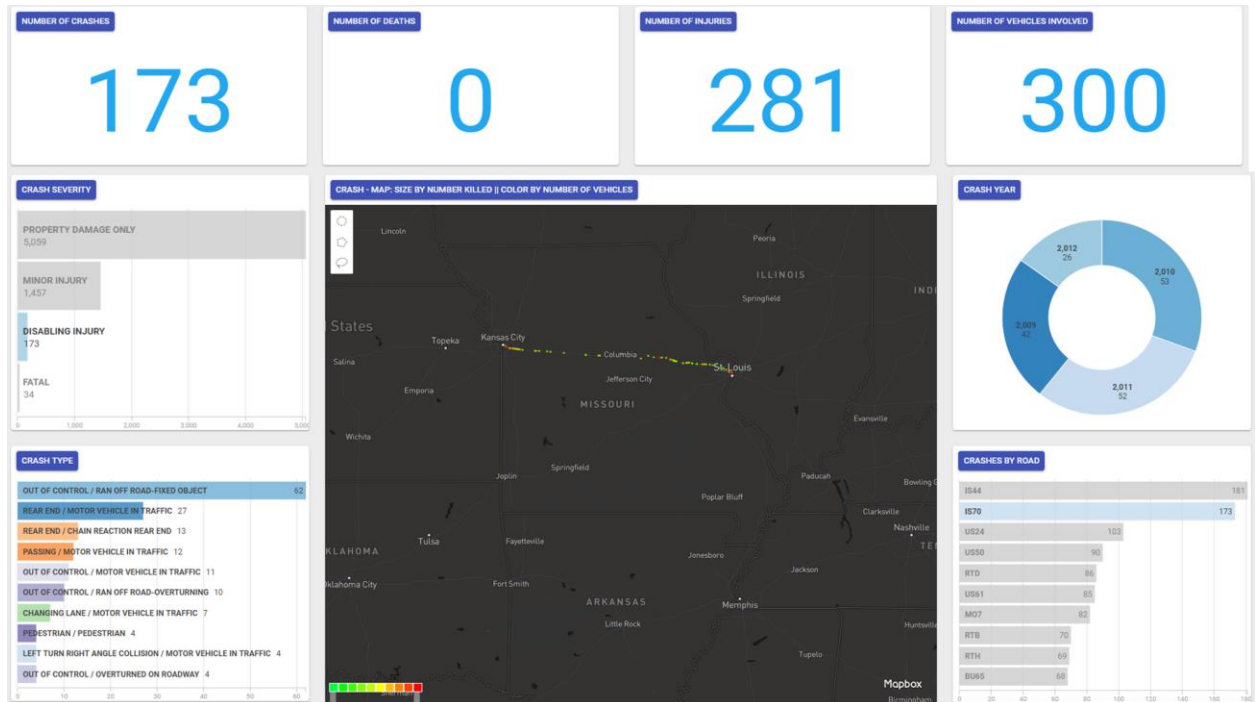
The following are examples of applications that utilize TITAN. They illustrate the capabilities of TITAN and serve to explain how it could be incorporated into a person's regular workflow. Figure 24 shows an example of a query of fatal right-angle crashes. The type of crash, right angle, is easily queried 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 were caused, and 50 vehicles were involved.



**Figure 24. Query – Number of Fatalities in 2012 Resulting from Right-Angled Crashes**

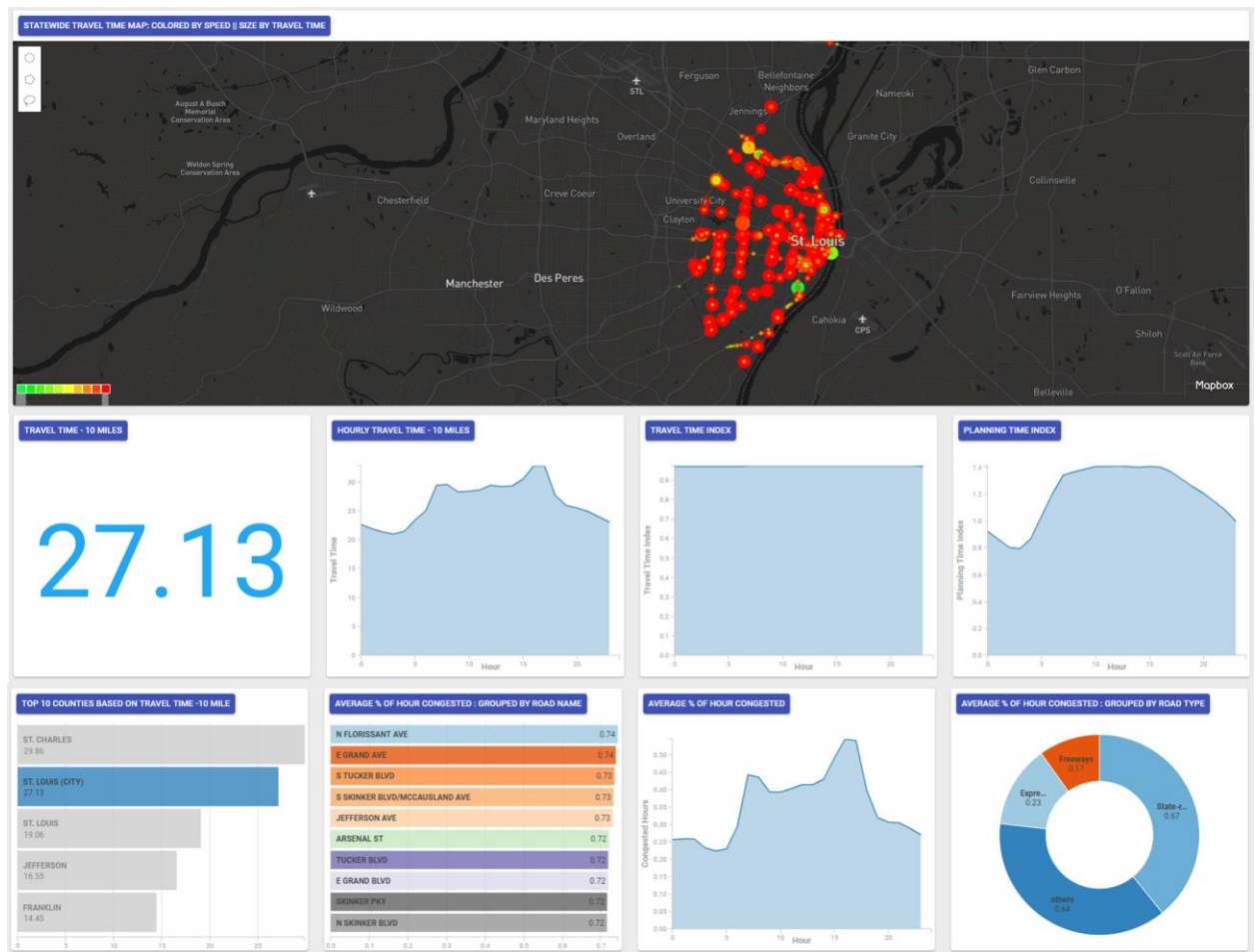
Figure 25 provides another safety example. The query is narrowed by severity

(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 39 is 281, and 300 vehicles were involved. Any of the narrowing factors can be selected easily by clicking on the corresponding label.



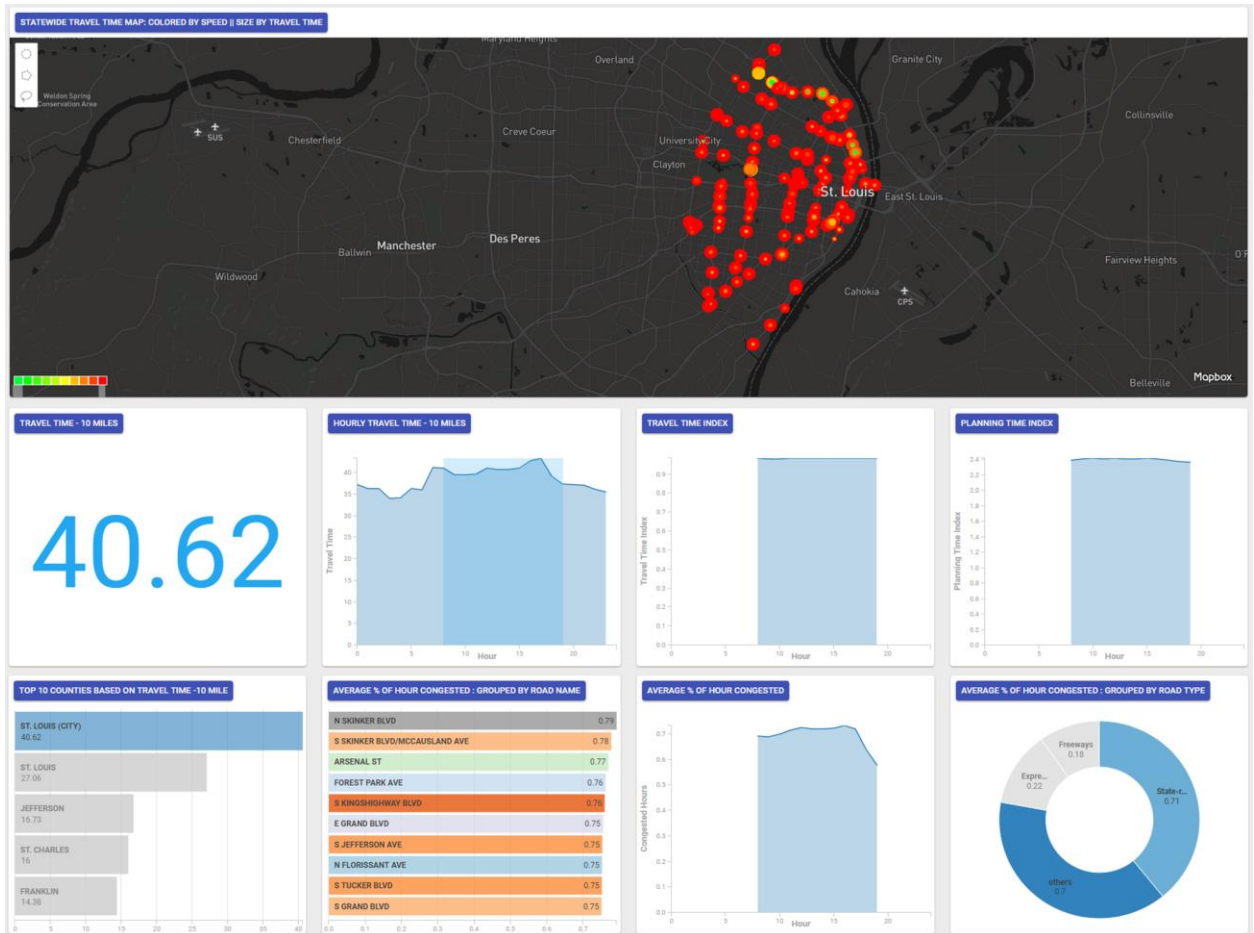
**Figure 25. Query – Number of Disabling Injuries on Interstate 70 Between 2009 and 2012**

Figure 26 illustrates 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 26. Query – Mobility trends in St Louis County**

Figure 27 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.



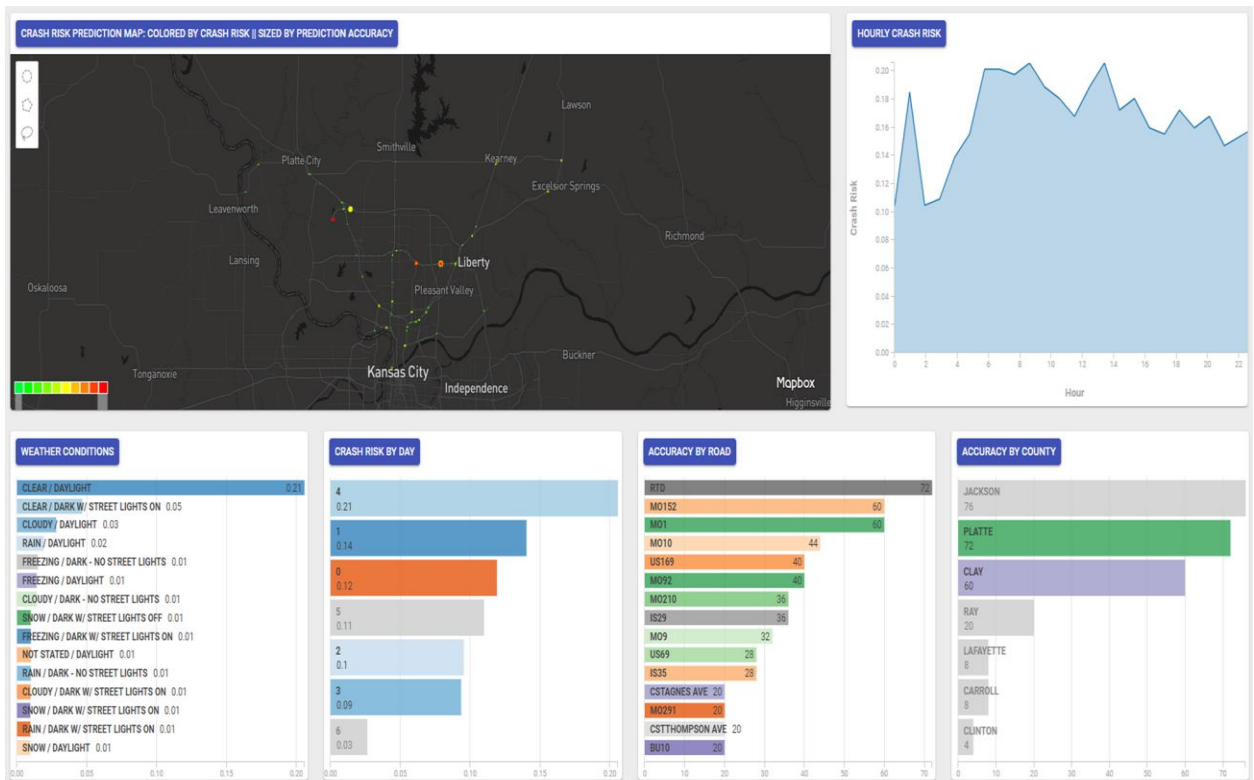
**Figure 27. Query – Mobility Trends in St Louis County between 8 am and 7 pm on State Routes**

Figure 28 shows a query on the opposite side of the state, in Jackson County, Kansas City. The query is of the am period only.



**Figure 28. Query – Crash Prediction in Jackson County between 6 am and 12 pm**

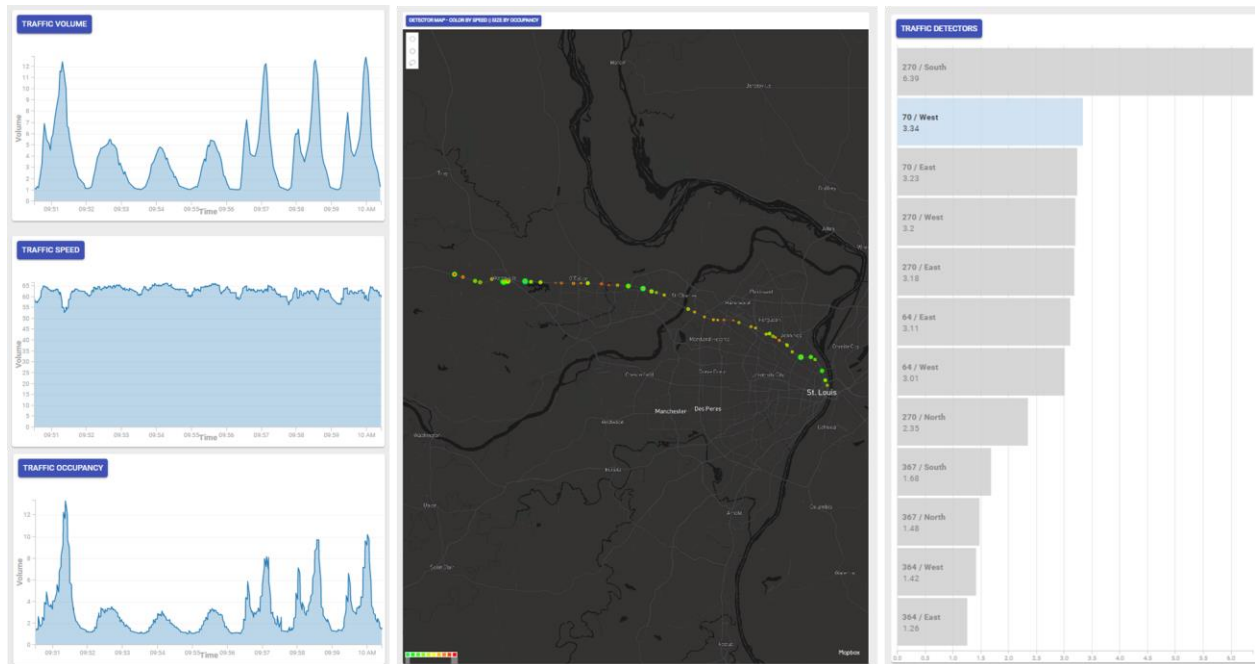
Figure 29 shows a query of multiple counties, i.e., Clay and Platte. The query also illustrates the display of weekdays only.



**Figure 29. Query – Crash Prediction in Clay and Platte County during Weekday**



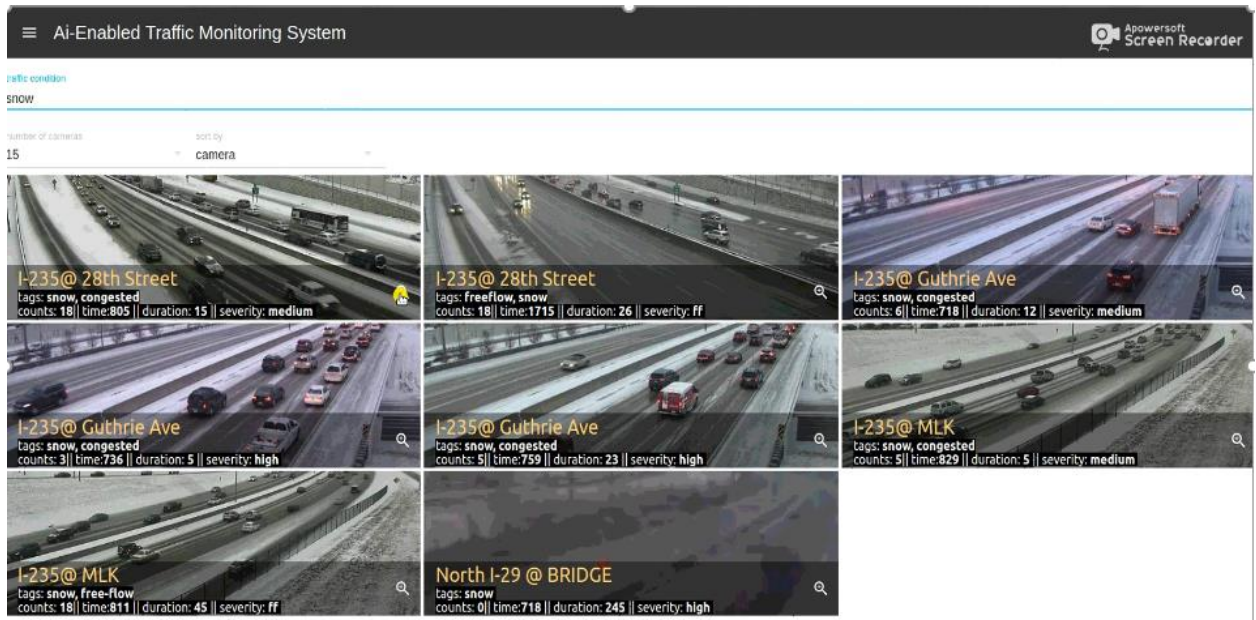
Figure 30 displays traffic detector data such as speed, occupancy, and volume. All three measures are displayed graphically on top of each other so that they can be easily compared.



**Figure 30. Query – Traffic Speed, Occupancy and Volume for Interstate – 70 West**

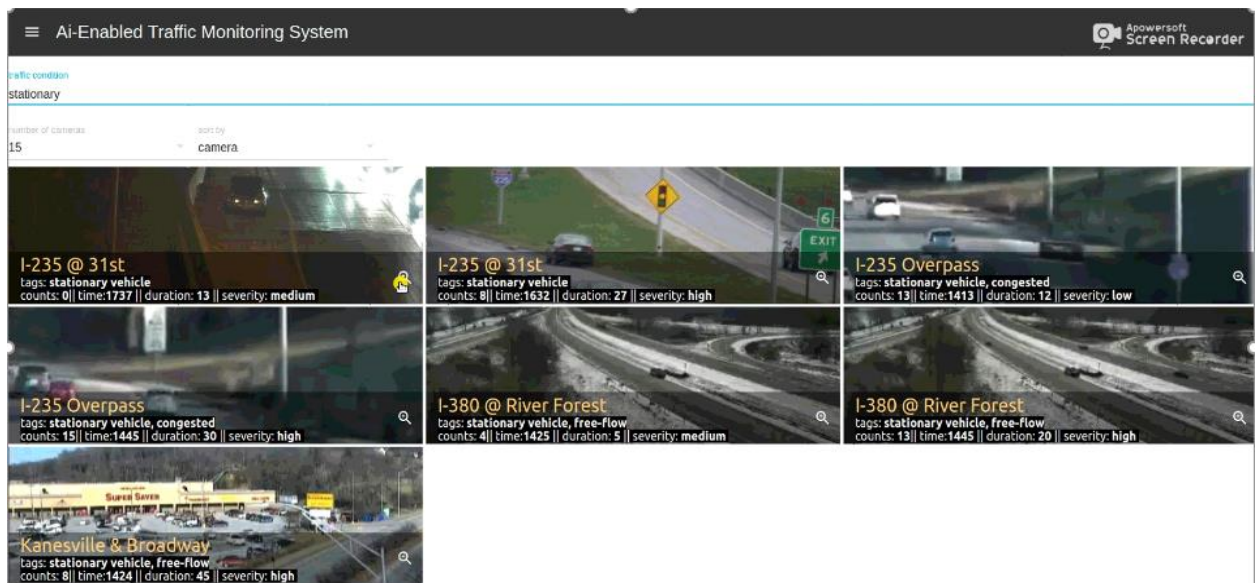
Figure 31 illustrates the machine learning (artificial intelligence) capabilities of TITAN. All camera views were processed with machine learning, and the roads with snow accumulation were automatically displayed. Such a query could be useful in winter weather to optimize response and improve traveler information.





**Figure 31. Query - Camera Locations with Snow on Ground**

Figure 32 again illustrates 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 improve traffic congestion due to rubbernecking.



**Figure 32. Query - Camera Locations with Stranded Vehicles**

## **CHAPTER 6: CONCLUSION AND FUTURE RESEARCH**

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 customizable 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, on the back end, a distributed cluster powered by Hadoop and Mongo DB.

TITAN has two main components: Data Center and Application 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 (application 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 discover insights quickly for decision-making. It was developed on top of a 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, etc.

For state DOT, TITAN has some practical advantages. First, the enormous amount of data being collected by DOT has great potential for improving data-driving decisions. However, such big data needs to be made accessible to various DOT staff who do not have the time and resources to dig into these massive databases. TITAN provides the tools to shrink down the effort required for data integration and analysis. Second, it provides graphical tools that simplify data querying and analysis. Querying involves simple steps such as drawing a circle around a region of interest, and then, it automatically produces related performance measures in various formats. Third, it 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 DOT employees.

Future updates and developments of TITAN will include data from other areas such as freight, pavement and bridge monitoring, pedestrians, and transportation performance evaluation.

## REFERENCES

1. Adu-Gyamfi, Y. O., Sharma, A., Knickerbocker, S., Hawkins, N. R., & Jackson, M. (2016). A comprehensive data driven evaluation of wide area probe data: Opportunities and challenges.
2. Adu-Gyamfi, Y. (2019). GPU-enabled visual analytics framework for big transportation datasets. *Journal of Big Data Analytics in Transportation*, 1(2), 147–159.
3. Amin-Naseri, M., Chakraborty, P., Sharma, A., Gilbert, S. B., & Hong, M. (2018). Evaluating the reliability, coverage, and added value of crowdsourced traffic incident reports from Waze. *Transportation Research Record*, 2672(43), 34–43.
4. Ayed, A. B., Halima, M. B., & Alimi, A. M. (2015, May). Big data analytics for logistics and transportation. In *2015 4th International Conference on Advanced Logistics and Transport (ICALT)* (pp. 311–316). IEEE.
5. Abbes, H., & Gargouri, F. (2016, January). Big data integration: A MongoDB database and modular ontologies-based approach. In *KES* (pp. 446–455).
6. Ali, S. M., Gupta, N., Nayak, G. K., & Lenka, R. K. (2016, December). Big data visualization: Tools and challenges. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)* (pp. 656–660). IEEE.
7. Abel, T. (2016). ReactJS: Become a professional in web app development.
8. Chodorow, K. (2013). *MongoDB: The definitive guide: Powerful and scalable data storage*. “O’Reilly Media, Inc.”
9. Chen, C. C., Knoblock, C. A., & Shahabi, C. (2006). Automatically conflating road vector data with orthoimagery. *GeoInformatica*, 10(4), 495-530
10. Dong, X. L., & Srivastava, D. (2013, April). Big data integration. In *2013 IEEE 29th international conference on data engineering (ICDE)* (pp. 1245–1248). IEEE.

11. Donalek, C., Djorgovski, S. G., Cioc, A., Wang, A., Zhang, J., Lawler, E., ... & Davidoff, S. (2014, October). Immersive and collaborative data visualization using virtual reality platforms. In *2014 IEEE International Conference on Big Data (Big Data)* (pp. 609–614). IEEE.
12. Guido, G., Rogano, D., Vitale, A., Astarita, V., & Festa, D. (2017, June). Big data for public transportation: A DSS framework. In *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* (pp. 872–877). IEEE.
13. Kadadi, A., Agrawal, R., Nyamful, C., & Atiq, R. (2014, October). Challenges of data integration and interoperability in big data. In *2014 IEEE International Conference on Big Data (Big Data)* (pp. 38–40). IEEE.
14. Keim, D., Qu, H., & Ma, K. L. (2013). Big data visualization. *IEEE Computer Graphics and Applications*, 33(4), 20–21.
15. Li, Z., Hodgson, M. E., & Li, W. (2018). A general-purpose framework for parallel processing of large-scale LiDAR data. *International Journal of Digital Earth*, 11(1), 26–47.
16. Malucelli, A., & da Costa Oliveira, E. (2003, November). Ontology-services to facilitate agents' interoperability. In *Pacific Rim International Workshop on Multi-Agents* (pp. 170-181). Springer, Berlin, Heidelberg.
17. Ma, X., Wu, Y. J., & Wang, Y. (2011). DRIVE Net: E-science transportation platform for data sharing, visualization, modeling, and analysis. *Transportation Research Record*, 2215(1), 37–49.
18. Mohammed, E. A., Far, B. H., & Naugler, C. (2014). Applications of the MapReduce programming framework to clinical big data analysis: Current landscape and future trends. *BioData Mining*, 7(1), 22.
19. Prasad, B. R., & Agarwal, S. (2014). Handling big data stream analytics using SAMOA framework-a practical experience. *Int. J. Database Theory and Applicat*, 7(4), 197–208.

20. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)* (pp. 1–10). IEEE.
21. Sun, C., Adu-Gyamfi, Y., & Edara, P. (2019). *TITAN – An Interactive Web-based Platform for Transportation Data InTegration and Analytics* (No. cmr 19-006).
22. Tanna, M., & Singh, H. (2018). *Serverless web applications with react and firebase: Develop real-time applications for web and mobile platforms*. Packt Publishing Ltd.
23. Vlahogianni, E. I. (2015). Computational intelligence and optimization for transportation big data: Challenges and opportunities. In *Engineering and Applied Sciences Optimization* (pp. 107–128). Springer, Cham.
24. Yuan, S., & Tao, C. (1999). Development of conflation components. *Proceedings of Geoinformatics*, 99, 1-13.
25. Zhu, L., Yu, F. R., Wang, Y., Ning, B., & Tang, T. (2018). Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1), 383–398.