

Research Article

SAVE-T: Safety Analysis Visualization and Evaluation Tool

Yuan Zhu¹, Sami Demiroglu², Kaan Ozbay³, Kun Xie⁴, Hong Yang⁵ and Di Sha⁶

¹Inner Mongolia Center for Transportation Research, Inner Mongolia University, Rm A357c, Transportation Building, Inner Mongolia University South Campus, 49 S Xilin Rd., Hohhot, Inner Mongolia 010020, China

²Agileassets, Inc., Austin, TX 78746, USA

³Department of Civil and Urban Engineering & Center for Urban Science and Progress (CUSP), Tandon School of Engineering, New York University (NYU), 15 MetroTech Center, 6th Floor, Brooklyn, NY 11201, USA

⁴Department of Civil & Environmental Engineering, Old Dominion University (ODU), 135 Kaufman Hall, Norfolk, VA 23529, USA

⁵Department of Computational Modeling and Simulation Engineering, Old Dominion University (ODU), 4700 Elkhorn Ave, Norfolk, VA 23529, USA

⁶C2SMART Center, Department of Civil and Urban Engineering, Tandon School of Engineering, New York University (NYU), 15 MetroTech Center, 6th Floor, Brooklyn, NY 11201, USA

Correspondence should be addressed to Yuan Zhu; zhuyuan@imu.edu.cn

Received 26 March 2021; Revised 23 June 2021; Accepted 12 July 2021; Published 23 July 2021

Academic Editor: Shamsunnahar Yasmin

Copyright © 2021 Yuan Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic crashes are one of the biggest issues which constitute a threat to lives of the motorists and disrupt operations of the transportation system. To reduce the number of crashes and alleviate their impacts, it is necessary to scrutinize the factors contributing to the risk of traffic crashes. Lately, visual analytics tools become very popular for data exploration and obtaining insights from the data. In this paper, a new web-based data visualization tool called Safety Analysis Visualization and Evaluation Tool (SAVE-T) was introduced. This tool enables users to interactively create queries and visually explore the results. By utilizing an online crash database, it offers various innovative functionalities for analysis and visualization of the crash data such as custom query development module and a subway-like map for easily visualizing the accident on the roadway segments. This tool provides an effective and efficient way to transportation agencies and professionals for traffic safety analyses and visualizations.

1. Introduction

Vehicle crashes, also known as traffic collisions or traffic accidents, are among the most significant issues for any transportation systems in the world. In 2016, road injury was ranked as the eighth cause of death in the world, and the number one cause of injuries, according to World Health Organization [1]. In the USA, vehicle crashes resulted in 37,461 deaths [2], more than 4.6 million injuries, and property damage totaling of \$432 billion [3] in 2016. In 1997, to emphasize that crashes are not accidental events and can be prevented, National Safety Council's Board of Directors declared to replace the word "accident" by "crash" to

describe traffic collisions [4]. However, effective tools are needed for obtaining insights from past crashes in order to prevent future ones. To achieve this goal and alleviate the impact of crashes, visual analytics tools become very popular for data exploration and obtaining insights from the data. This tool also has the potential to be beneficial for conveniently exploring historical crash data and analyzing the contributing factors.

New Jersey Turnpike (NJTP) and Garden State Parkway (GSP) are two of the busiest highways in the United States [5]. These two toll roads are maintained by the state agency called New Jersey Turnpike Authority (NJTA). On both roadways, traffic crashes remain as one of the critical issues

that need to be analyzed using crash data from New Jersey Department of Transportation (NJDOT) [6] by using customized reports.

A web-based visualization and analytics tool, named SAVE-T, that is designed to work with NJ's crash database in general, was developed by the authors of this paper and was tested by NJTA. This user-friendly tool is aimed at improving productivity of engineers by automating processing and report generation from the crash and other traffic data. Many features, which make it easier to access, store, manage, visualize, and update crash and other related traffic data, are implemented in the tool based on the user needs. Moreover, many customizations to the tool were made to better fit the tool to the needs of state troopers and other prospective users. This tool features the following properties:

- (i) Better productivity: the tool uses JavaScript, PHP, and AJAX to create the SQL queries on the fly and send query criteria to a database and retrieve results. All the joins and queries are executed at server side, and hence the execution speed is independent of the resources on the client side. The tool is very flexible and supports a variety of query types, including map queries, time frame queries, and customized queries consisting of user-defined factor(s). These functions can be achieved through the user interface (UI), without requiring manually written queries.
- (ii) Multilevel user hierarchy: the tool supports different user types (expert, regular, and trooper) which have access to various features and privileges inside the tool.
- (iii) A novel and interactive way of data visualization, including active demonstration of roadway segments, line map-based visualizations, and heatmap queries: One of the innovations of this tool was the use of a simple line map instead of a complex GIS interface which makes it difficult to maintain and cumbersome to run the app on mobile platforms. Results are visualized using on-screen tables/charts and they can be exported in MS Excel, PDF, and a printer-friendly format.
- (iv) No expert knowledge is required: users can select the crash-related parameter directly on the UI for preparing visual queries, rather than writing cryptic scripts. In the back-end of the tool, the necessary code is compiled at the runtime. Upon execution of the query, its script can be also exported and used/modified in other database tools. Hence, the expert user has the flexibility to fine-tune the queries based on their need and continue to use them with their tool-of-choice.
- (v) Cross-platform compatibility: the tool was developed using up-to-date programming languages and packages, which enables the tool to run on web browsers of portable devices.

2. Literature Review

In recent years, many safety scientists and agencies developed computerized tools for the analysis of traffic safety data. While some of these tools are standalone software packages, the others utilize web-based mapping APIs (Application Programming Interface) for their Geographic Information Systems (GIS) needs at national, state, and local levels.

One of the earliest attempts is the exploration of highway design using a software called the Highway Geometric Design Consistency Evaluation Software developed by Krammes et al. [7]. This is a menu-driven program for evaluating the design of rural highways using preliminary models. SafetyAnalyst [8] is the most established and state-of-art traffic safety analysis toolset developed by AASHTO. SafetyAnalyst implements analytical procedures to identify and manage the safety improvement measures for a highway by cost-effective means. The tool automates procedures by implementing six main steps of highway safety management process, including network screening, diagnosis, countermeasure selection, economic appraisal, priority ranking, and countermeasure evaluation. However, its features can be too limited for the specific needs of the agencies, and the customized features cannot be easily implemented in its user interface.

Among the web-based traffic safety tools, the most well-known is Transportation Injury Mapping System (TIMS) [9], which is developed by SafeTREC Research Center in University of California Berkeley. The tool provides a way to access the Statewide Integrated Traffic Records System (SWITRS), where the crash data of California is maintained. The tool has various features for querying and mapping the data. The TIMS also supports safety performance management, collision diagram and hotspot analysis, and other GIS tools for different crash types. However, the TIMS is a state-level safety analysis and visualization tool developed for research purposes only; hence, the operational needs of the transportation agencies were not taken into account in its development.

On the other hand, for the development of the tool in this paper, operational needs of the local authority are taken into account and facility-specific features of both roadways are visualized in the tool. In addition, general rule-of-thumb in the software tool which is good interoperability and code reusability is also followed. These considerations together with the current state of the art have formed the foundation of this tool.

3. Data

The proposed tool utilizes the NJDOT's crash data which is publicly available at [6]. To create this database, the crash reports, also known as State of New Jersey Police Crash Investigation Report NJTR-1 [10], are filled out by New Jersey State Police for investigation of any motor vehicle crashes on NJ highways. In this paper, we focused on the crashes on NJTP and GSP to better focus on the capabilities of the developed tool. It must be noted that the format of NJTR-1 report complies with the national

standard: Classification of Motor Vehicle Traffic Accidents (ANSI D16.1-2007) [11]. A sample NJTR-1 report is shown in Figure 1. These reports contain detailed information on the crash as well as vehicles and occupants. The data in these reports is also entered into their online system by the officers in the field. Upon submission of a report to the system, it also becomes available to NJTA. The crash database was designed based on NJTR-1 coding protocols, which focuses on three entities: crashes, vehicles, and vehicle occupants. Hence, the crash database also contains three tables:

- (1) "Accident" table has general information related to the vehicle crashes,
- (2) "Accident_Vehicles" table includes detailed information of each vehicle involved in the collisions.
- (3) "Accident_Vehicle_Occupants" table contains the information about the vehicle occupants (either drivers or passengers) involved in the accidents.

Any changes in NJTR-1 directly affect the structure of the crash database. In 2017, major revisions were made to NJTR-1 involving the addition of new values to certain fields of the crash database. For example, new values were added to distraction due to mobile devices field and safety restraints field was expanded with two new categories for child restraints. There were also changes in the environmental factors, roadway conditions, and driving under influence (DUI) fields. Tackling all these issues require development of a tool that is highly backward and forward compatible with the changes to the data structure.

Figure 2 briefly presents the crash statistics from 2013 to 2017, which are retrieved from the crash database. These tables are related to each other by *Acc_case* (Crash ID) field; hence it is possible to combine different parts of the data by joining multiple tables by this field. Since SAVE-T can access the up-to-date database in real-time, join-type queries are compiled at the runtime of the tool without requiring any additional processing of the data.

4. Architecture and Functionalities of SAVE-T

4.1. Software Architecture. The architecture of this application consists of three parts: Remote servers, mirrored (local) server, and clients. As a design choice, Linux-based machines that are connected to the Internet are used for hosting the application. Due to the layered security features of the app, the copy of tool that is running on the development server, outside the authority network, only has access to the mirrored database. Hence, daily backups of the database are taken by means of an automated Cron job. In this way, the new records, if existing, in the crash database on the remote server are downloaded to the local test server. Finally, based on the results of the check, the mirror database is updated as necessary. While updating the records in the database, by means of the special embedded routines, the tool also parses manually entered text fields, such as damage to Turnpike property field, so that they can be later used in the analyses.

The development version of the tool and mirror database are currently running at a dedicated server located at NYU. Since the tool is written mostly in PHP, an Apache webserver also runs at the same machine. The clients can connect to the tools' interface using their web browsers. The client-side functions of the tool are developed in JavaScript and server-client interactions and data transfer requests from the server utilize AJAX and JavaScript Object Notation (JSON).

Most of the data processing and report generation related function of the tool run at the Linux-based server. For data storage and management needs of the tool, both MySQL and Oracle database platforms are utilized, while MySQL is used for general data storage needs of the tool, which is a widely used freeware.

The UI of the app uses the latest technologies including PHP, JavaScript (JS), JQuery, AJAX, and d3.js, which can respond seamlessly to users' input without refreshing the interface every time. The web-based GIS tool Leaflet.js are used to implement the segment map on the client side. This makes it visually attractive and easy to use without delays that might be experienced by some other database tools. The proposed tool is secured so only authorized users with the correct credentials have access to its interface. When a user login to the app, at first, the user is welcomed by user controls such as dropdown boxes, input boxes that define the parameters of the queries. There is also a simple subway-like line map that helps to define the mileposts/interchanges ranges of the roadways included in the query. Initially, the results of any query are presented in the form of on-screen tables and charts. Each query run by the user is also saved and the users have access to their list of previous actions in the app. This feature can be used for creating a report for all or selected queries from the list. Moreover, there are many report formats available in the tool including Excel, Word, and PDF.

4.1.1. Model-View-Control (MVC) Structure. SAVE-T adopts a Model-View-Controller (MVC) structure as seen in Figure 3. The model is the key module that interacts with the database. While receiving the SQL query request, after internally interpreting it, it runs the query at the database and monitors the query execution. Then the model receives the query results from the database. The model also connects to the buffer file which is also used for storing the previous frequent queries.

The view is the front-end of the tool where the user interacts with the server back-end of the tool. The UI supports multiple types of input formats for query conditions and output rendering options including result tables, graph, and charts and exporting of these results. The view uses JSON as both input and output format.

The controller is the module that transmits information between model and view. Upon receiving query conditions from the view, the controller translates these conditions into SQL query script and sends the executable script to the model. After receiving the query results from the model, it encodes the query results to JSON format and sends them back to the view. The controller is also responsible for saving

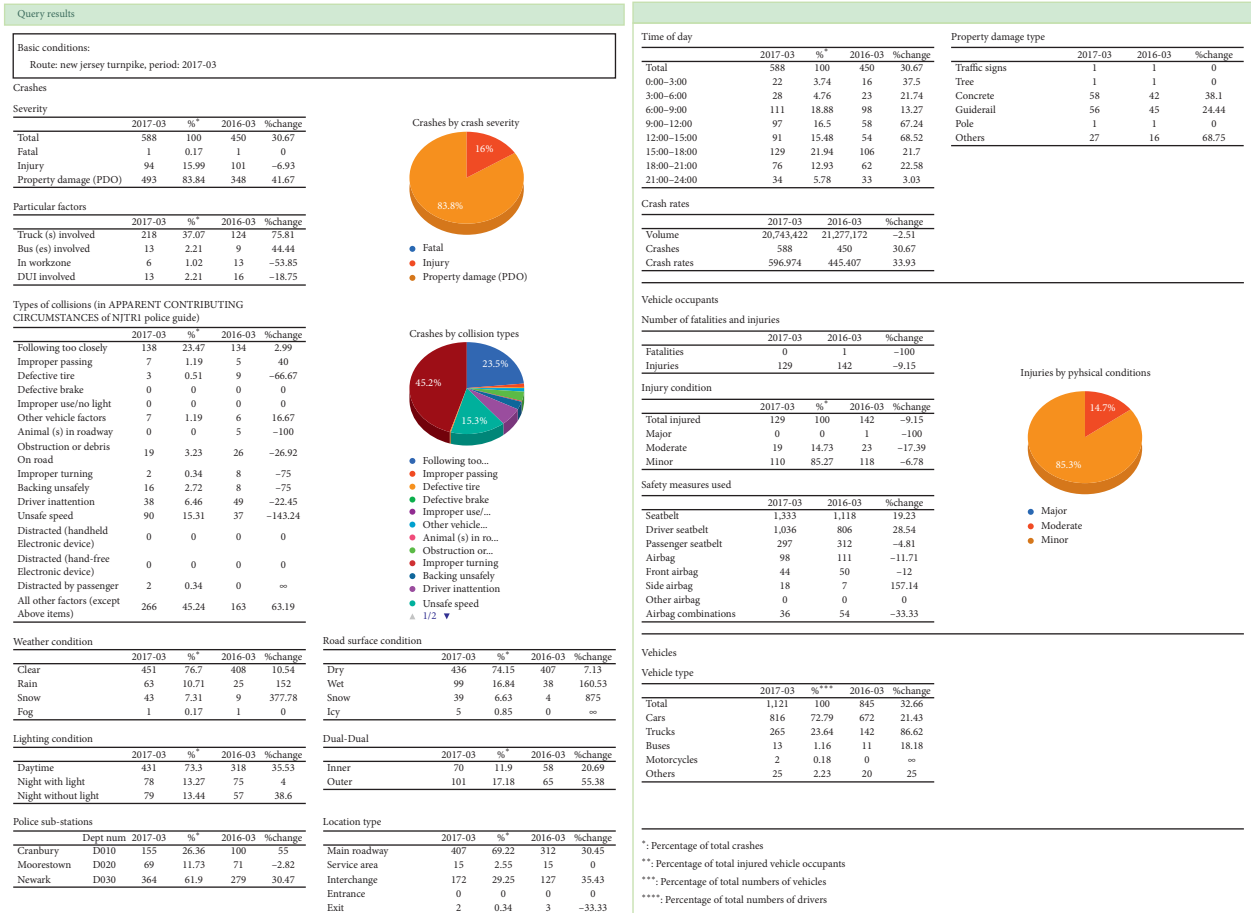


FIGURE 1: Crash report of the tool (March 2017 data).

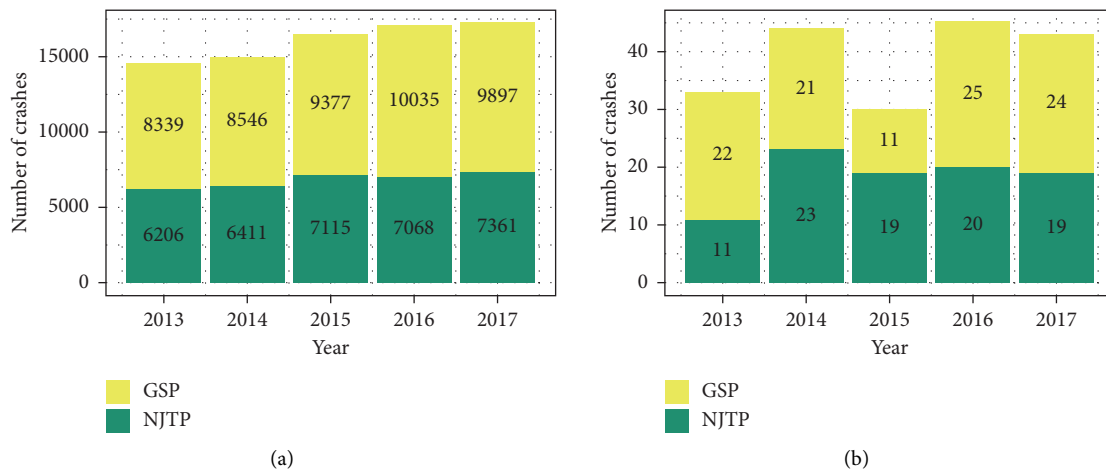


FIGURE 2: Continued.

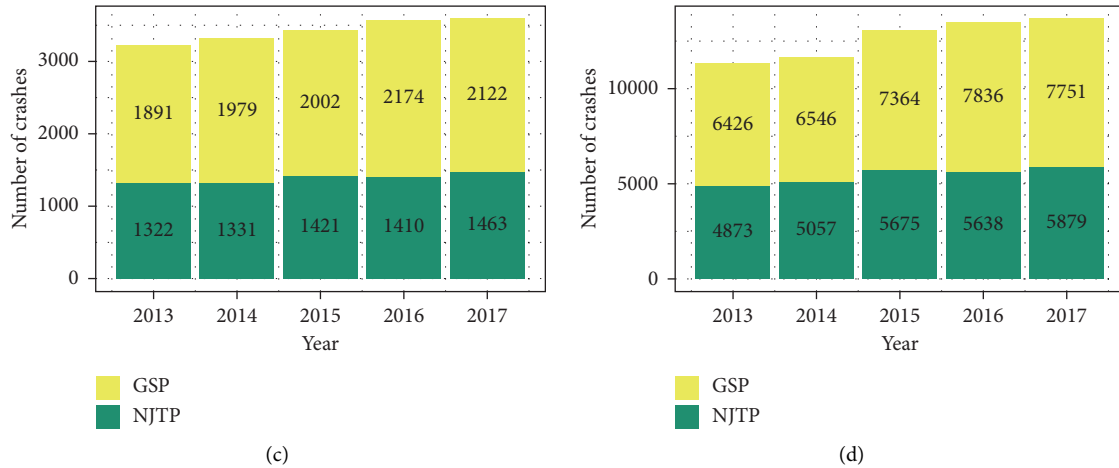


FIGURE 2: Number of crashes by years. (a) Total crashes, (b) fatal crashes, (c) injury crashes, and (d) PDO crashes.

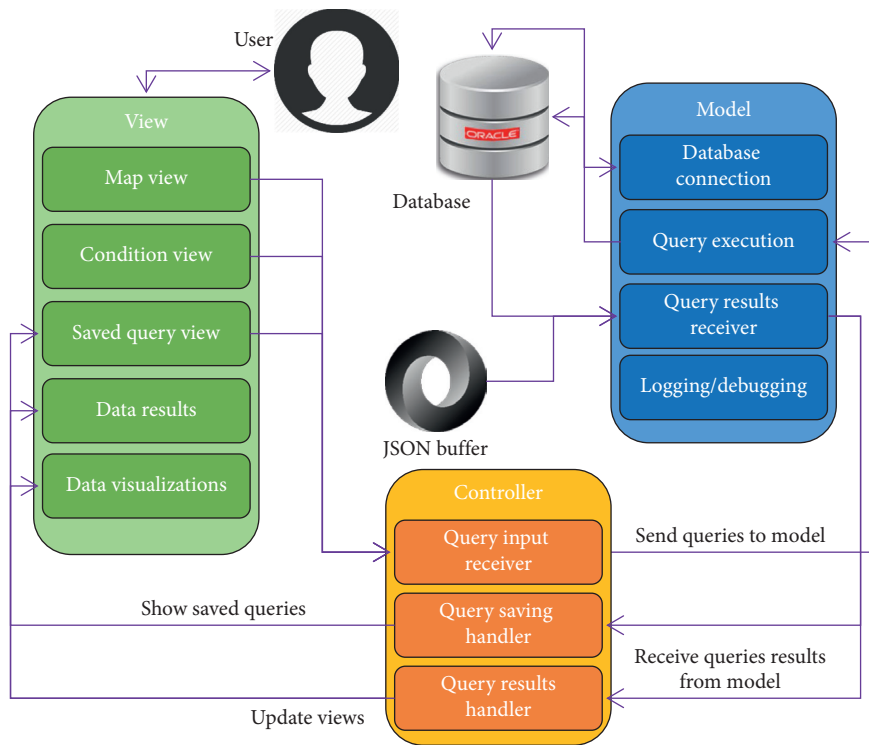


FIGURE 3: MVC structure of the app.

the query scripts and results into its JSON buffer. Model, view, and controller are encapsulated separately in SAVE-T. This gives the tool better scalability and interoperability.

Based on the MVC structure, the app set up processes that occur between the front-end interface and back-end services of the tool. There are five modules in the main interface: login and user control, condition query, map query, query export, and result visualization. Upon successful login, the index page shows the dashboard and query options of the tool to the user. After the submission of a query, the modules on the server side handle this request from the index page and pass its conditions to JS

middleware; then these conditions are converted to a SQL query and send back to the PHP functions at the server using AJAX. After retrieval of the query results from the database, the back-end PHP functions send results back to JS middleware and finally generate result visualizations on the UI.

4.2. Functionalities of the Tool

4.2.1. User Control. Users can access the main interface of the tool using their username and password. For security reasons, the password of each user is encrypted and saved as MD5 strings in the MySQL database. Currently, there are

three types of users in the tool. The experts can access general and detailed queries and manage other users, such as adding or deleting users and changing user types and passwords, etc. The common users can only access the general query and change their user passwords. The third user type is called the state trooper. Upon logging in, these users are redirected to their special interface tailored for the specific needs of state troopers.

4.2.2. Query Types in the Tool. SAVE-T supports multiple options for defining the query conditions. General queries provide a convenient and easy way to generate a report for the whole system with a single click, which is accessible to all types of users. The detailed queries enhance the flexibility of the tool by letting the users define their own custom constraints in the query. It should be noted that this feature is only available to expert users and state troopers. On the other hand, state troopers also get to their own features such as the use of segment map or heatmap which saves time in defining the query conditions.

General Query. General queries are designed for generating the crash reports for a particular month or for a whole year. For this query, there are simply three combo boxes for selecting the query conditions: query of year, month, and roadway. If month combo box is set to “Annual,” the query of the entire selected year is executed. This query is executed for the entire length of the selected roadway from combo box such as NJTP, GSP, or both. Since this query is used for generating routine safety reports, there are also some simple optimizations embedded inside the tool. This includes buffering of the query results in JSON format when they run the first time and then retrieving them if/when accessed again in the future.

Detailed Query. The canned query has a specific use case, and therefore it lacks a flexible mechanism for selecting other crash-related conditions. However, this feature is complemented by the detailed query. Following parameters in the detailed query can be customized:

- (1) Date period: instead of using only year and month, date period can be selected as any start and end date using the date pickers. The users can also choose customized dates for the comparison period. By default, the same dates in the previous year are automatically selected by the tool as the comparison period.
- (2) Roadway: this helps in selecting the entire length or a partial segment of the roadway. If “Entire Route” checkbox is disabled in the tool, this means that other roadway segment-related conditions such as milepost range, directions, branch, and spurs become available, and the user can select these constraints by using the sliders or manually input them.
- (3) Specific factors: this is an optional feature which enables users to select additional constraints to the query up to 24 collision-related factors and three

other factors related to the vehicle occupants. It should be noted that any combinations of these factors are available to the users for developing customized queries.

Milepost (MP) Level Query. This option is activated in the tool only if a roadway is selected and “With Milepost Report” checkbox is checked. When selected this query generated an MP report in addition to the general report of the tool. Please check Table 1 and the following chapter for the details of the MP report feature of the tool.

Segment Map and Heatmap Query. The segment map and heatmap query are available in the separate dashboard-style UI specially designed for the NJ state troopers. In this module, the roadway segments of the expressways are visualized in subway-like map using d3.js and on the map using Leaflet.js. The subway map and the actual map are interconnected and highlighted when a specific segment is selected. Upon selection, heatmap of statistics for up to 12 months can be visualized. The details of these queries are described later in the paper.

4.2.3. Canned Crash Reports. SAVE-T can currently generate up to three types of predefined canned reports: General report, MP-based reports, and an incident responder specific report.

General report can be created for both general queries and detailed queries. It should be noted that the resulting charts and tables of this type of query are specially designed based on the needs and practices of the agency using the tool. This report was basically comparing the frequency of the various types of crashes for the moth month under consideration versus the previous year’s same month. The format of the general report follows the very same format of the crash reports previously adopted by the agency. The results of this type of report can be exported directly into an Excel sheet, which is identical to the format previously manually generated by the safety experts at the authority. In the development phase of this feature, considerable labor was spent on the development and testing of the complex queries, which required joining all crash-related data from the crash tables.

MP-based report offers optional results generated in-demand and consists of a table showing the milepost to milepost crash frequency table for the selected segment of roadway. For this report, the crashes are also categorized by the type of the road-side feature such as interchange, entrance and exit, service areas, and inner and outer roadway (for the divided sections of the roadways). Based on the user feedback, an option to filter out to the interchange crashed is also implemented in the tool.

Incident response specific report provides a customized version of the general report that is designed specifically for the needs of incident responders. Based on the needs of the incident responders, some key criteria are defined, such as the number of crashes by severity and jurisdictions, DUI involved crashes and DUI test results, usage of safety

TABLE 1: Overview of query constraints based on NJTR-1 report.

Query type	Subcategory	Constraint	Values	Remarks
Date periods	Query period Period for comparison	Start and end dates	(Selected from Datpicker)	— Optional
		Mainline undivided	MP: 0–48.8	Direction: NB, SB
Roadway	NJTP	Mainline dual roadway	MP: 48.8–105.8	Direction: NB, SB, dual: inner, outer
		East spur and west spur	MP: 105.8–117.2	Direction: NB, SB, dual: inner, outer
		Newark bay extension	MP: 0–8.2	Direction: EB, WB
		Penn Turnpike extension	MP: 0–6.7	Direction: EB, WB
		GSP	Mainline	MP: 0–172.4
Specific factors	Collision factors	Day of week	Su, Mo, Tu, We, Th, Fr, Sa	
		Time of day	Eight even-distributed time periods from 0:00 to 24:00	
		Apparent contributing factors	23 contributing factor codes in NJTR-1	
		Weather	10 weather conditions codes in NJTR-1	
		Number of injuries	0, 1, 2, 3, 3+	
		Number of killed	0, 1, 2, 3, 3+	
		Location type	Main roadway, service area, interchange, entrance, exit	Checkbox
		Lighting conditions	7 lighting condition codes in NJTR-1	
		Police substations	Moorestown, Cranbury, Newark, Bloomfield, Holmdel, Galloway (Bass river)	
		Property damage type	Traffic sign, tree, concrete barrier, guiderail, pole, others	
		Road conditions	9 road surface type codes in NJTR-1	
		Severity	Fatal, injury, property damage only	
		Type of collisions	13 collision type codes in NJTR-1	
		Dual-dual	Inner, outer	Checkbox, for dual roadways
		Truck involvement	Yes/no	
Bus involvement	Yes/no			
Motorcycle involvement	Yes/no	Radio button		
Hit and run	Yes/no			
Alcohol involvement	Yes/no			
Degree of injury	Major, moderate, minor			
Factors of vehicle occupants	Safety equipment use	12 codes in NJTR-1 regarding usage of belts, harness, child restraint, helmets, and airbag	Checkbox	
		Ejected	Not ejected, partial ejection, ejected and trapped	
MP level query	NJTP	Mainline undivided		Direction: NB, SB
		Mainline dual roadway	Total, interchange, service area	Direction: NB, SB, dual: inner, outer
		East spur and west spur		Direction: NB, SB, dual: inner, outer
		Newark bay extension	Total	Direction: EB, WB
		Penn Turnpike extension	Total, interchange	Direction: EB, WB
GSP	Mainline	Total, inner, outer, entrance, exit, lot	Direction: NB, SB	

measurement, crash causes, etc. This report type has its own compact interface designed based on the aforementioned key criteria.

Both general report and MP-based reports of crash results can be exported to a spreadsheet, PDF, or a printing-friendly format. The tool also supports query export, which

saves the most recently executed query into a text file and makes it possible to verify the query on a database manager. A demo video of the SAVE-T can be found on <http://bit.ly/2Ywozeh>.

In the UI, there is also an option to investigate the query script of the generated report which is called “under the

hood.” When the user clicks to see the details of their past activities from action history, the specific query and its details are sent to the client side as JSON data. Then, under the hood option gives user ability to reach the raw SQL query script generated by the tool. In this part of the tool, the steps in the query are explained such as table(s) contained in the query, conditions in the query, and the result of the query. It also shows the selected fields and tables in the database.

5. Case Study: Data Analysis and Visualization

This section presents the functionalities of SAVE-T by means of a case study using NJDOT crash data of year 2017.

5.1. Crash Analytics. Gaining additional insights on the crash-related factors can be beneficial to any safety expert. The rest of this section discusses how the tool brings some light on the crash-related factors by carefully selecting the key factors and finally visualizing crashes by means of various charts and graphs. Figure 1 is generated using the crash data for March 2017 and shown alongside the data for the same period for the previous year. It can be observed from the figure that the number of crashes that occurred on NJTP is thirty percent more than the previous year. In both periods, there is only one fatal crash, and injury crashes are sixteen percent of total crashes.

Among all circumstances, following too closely is the most common cause of crashes that makes a quarter of all crashes. The second most significant type of collision is unsafe speed, which increased dramatically compared to the previous year. Based on weather and road surface conditions, the snow-related crashes increased in 2017, which implies the seasonal factors can be in the play. Based on the particular factors, truck-related crashes increased by 75% in 2017. This can be confirmed by the fact that the number of crashes in the outer lanes of the roadway, which is only segment trucks permitted to use for the divided part of the roadway, is also increased. From the location type table, it can be observed that nearly seventy percent of the crashes with known locations occurred on the main roadway. Time of day results reveal more insight into the temporal distribution of crashes. More crashes occurred on AM and PM peak periods than any other time period, as expected. Property damage types show that the concrete and guidrails are more susceptible to the crash than any other properties of the authority.

5.2. Vehicles and Their Occupants. There are three factors related to vehicle occupants in the crash report: Number of fatalities/injuries, injury conditions of the occupants, and safety measurement used by them. Based on the results shown in Figure 1, there are one fatality and 129 injuries in March 2017, which is close to the results in previous years. Among those injuries, 110 of them are minor injuries, while there are 19 moderate injuries. Seatbelt and airbag use are combined in the tool under safety equipment use section.

5.3. Milepost Query. The results of the milepost-based query are presented in Figure 4. In this figure, there are five series showing different types of crashes: total, northbound southbound of the roadway, interchange, and service area crashes. It can be observed that the number of crashes increases dramatically after milepost 80. This is in agreement with the characteristics of NJTP since this busier part of the roadway is in the vicinity of a densely populated area close to New York City.

5.4. Segment Visualization and Heatmap Query. As mentioned before, the state trooper users have access to another visualization called the segment maps. For example, suppose that the user is interested in the crash report between Exit 10 and Exit 11 in March 2017. As shown in Figure 5, after logging in, the link segment of the expressway is visualized as both subway style and actual map. On the subway style map, triggers are set up so that the information such as segment location and milepost can be accessed by hovering the mouse on top of the map. In addition, the roadway links inside the map are clickable, and numbers of crashes in the past twelve months are shown in the heatmap. It should be noted that when the link on the subway map is selected, it turns to red, and the corresponding segment on the actual map is highlighted; then the heatmap result only contains selected links. If no links are selected, the heatmap shows all sections on the roadways, and the map shows the location of the entire expressway.

On top of the heatmap, there are three combo boxes. The first one is used to reorder the heatmap by row or by column based on the key values (in this case number of crashes). Color palettes can be adjusted using the second combo box. By changing the value of the query type combo box, the heatmap can visualize additional dataset as crash rate, number of fatalities, etc.

Each cell shows the number of crashes within designated interchanges and month. By clicking the cell of Exit 10 to 11 in March 2017, relevant constraints are applied and executed, and query results of this milepost range and month are generated. In this case in Figure 5, when hovering on the designated cell of heatmap, the result shows that there are 11 crashes in March 2017. When the cell is clicked, the tool identifies the milepost range and time period for the query based on the selected segment and applies the constraints to lower slider and datepicker and executes the query instantaneously by itself. According to the results, there are two injury crashes and nine property damage only (PDO) crashes. Only one crash occurred in rainy conditions for the selected period. Moreover, eight crashes occurred at interchanges, where six are at Exit 10 and two are at Exit 11.

In addition, the volume data are available for NJTP between 2013 and 2017, which enable the calculation of the crash rate of each link on NJTP, as shown in Figure 6. Crash rates are calculated as crashes per million vehicle-mile-traveled (VMT):

$$Cr = \frac{c_{total}}{(Vol/10^6) * LinkLength}, \quad (1)$$

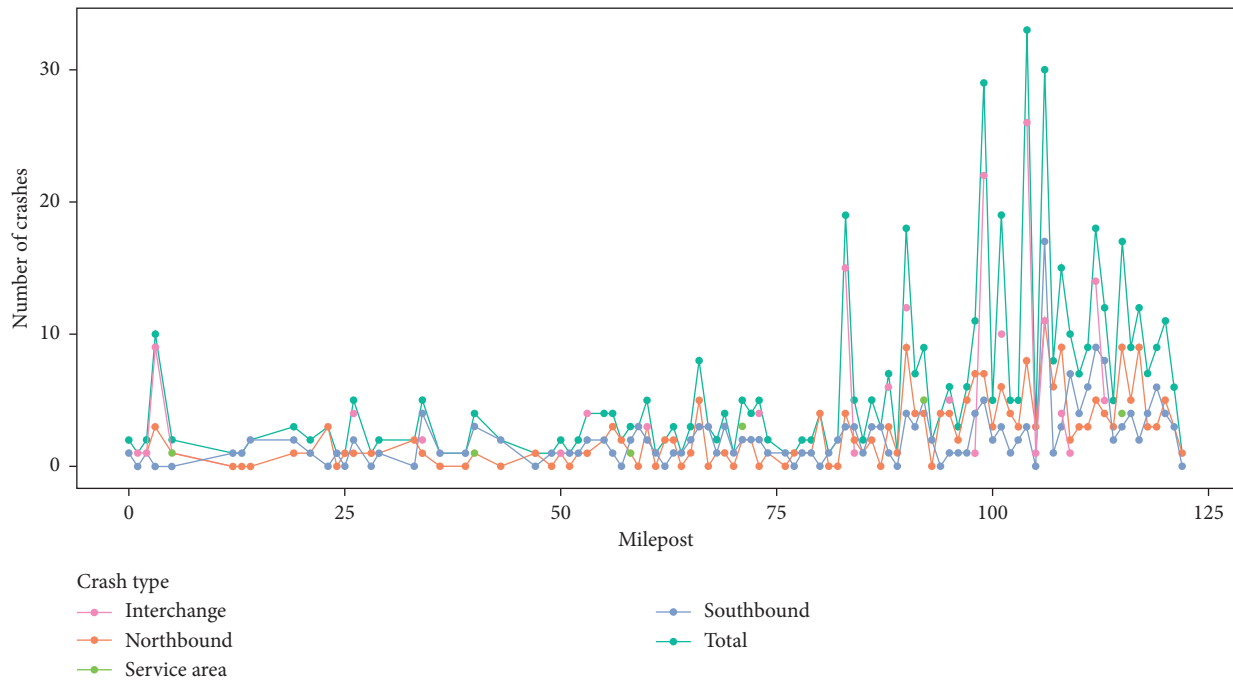


FIGURE 4: Number of crashes in terms of NJTP mileposts (March 2017).

where Cr is the crash rate, c_{total} is the total number of crashes on the roadway, Vol is the volume for an interchange-to-interchange segment on the roadway, and link length is the length of that segment. It is noticeable that the section of roadway between interchanges 14 and 14C coincides with the Newark Bay Extension of NJTP serving as a critical connector between NJ and New York City.

5.5. Performance of the Queries. In this section, speed tests are conducted to test the query performance of the tool. For each query type, ten runs are made, and average execution time is recorded. The reader should note that each time a query is executed, four subqueries are also performed. Two of them are the selected time period and the previous comparison period. The other two are the queries of crashes in the whole year of selection and the previous year. These annual queries run behind at the server to later facilitate the creations of spreadsheet and PDF reports. If any of the four was previously performed, the corresponding results are obtained from the buffer. For testing the query performance, the average time from the database is calculated after erasing web browsers' cache so that the results are generated from scratch instead of being fetched from the buffer.

As shown in Table 2, the query for all crash records in 2017 on both roadways almost takes ten seconds due to the size of the data retrieved as a result of the query. Annual query for GSP only takes six seconds, and monthly query takes even less than that. On the other hand, the execution times from buffer are nearly instantaneous for any of the three query types. For detailed queries, the performance is adversely affected by the complexity of the selected roadway segment and the number of selected crash factors. The query

performance can also be improved if the fields used in the query are indexed in the database, but since the app is not allowed to alter the production crash database, performance optimization is not considered for the tool. Query complexity is also the reason why the execution time of the road condition query takes more than six seconds. In addition, selecting MP level query, the query only increases execution time from 5.8 seconds to 6.2 seconds, which implies that the MP level query has very low overhead.

5.6. Approaches on Handling Difficulties between Tool Results and Crash Records. During the development process, some inconsistencies were observed between the results from SAVE-T and the actual crash statistics. The two key issues were as follows:

- (1) Sum of fatal, injury, and PDO crashes was different than the total number of crashes for the selected period
- (2) Some of the crash metrics generated in the reports were less than the actual values

Careful inspection of the dataset helped to understand the causes that contribute to these inconsistencies. For the first issue, it was found that, in the accident table, the crash records were not always unique. Querying the database for the unique records revealed that 2.3% of the crashes had more than one record. This stemmed from the updates to the crash records after the fact. For example, if there is an injured victim in the crash and later if the same victim dies due to the injuries, the crash record for this specific crash was needed to be updated resulting in creation of a duplicate record in the database. If these cases are ignored, the crashes

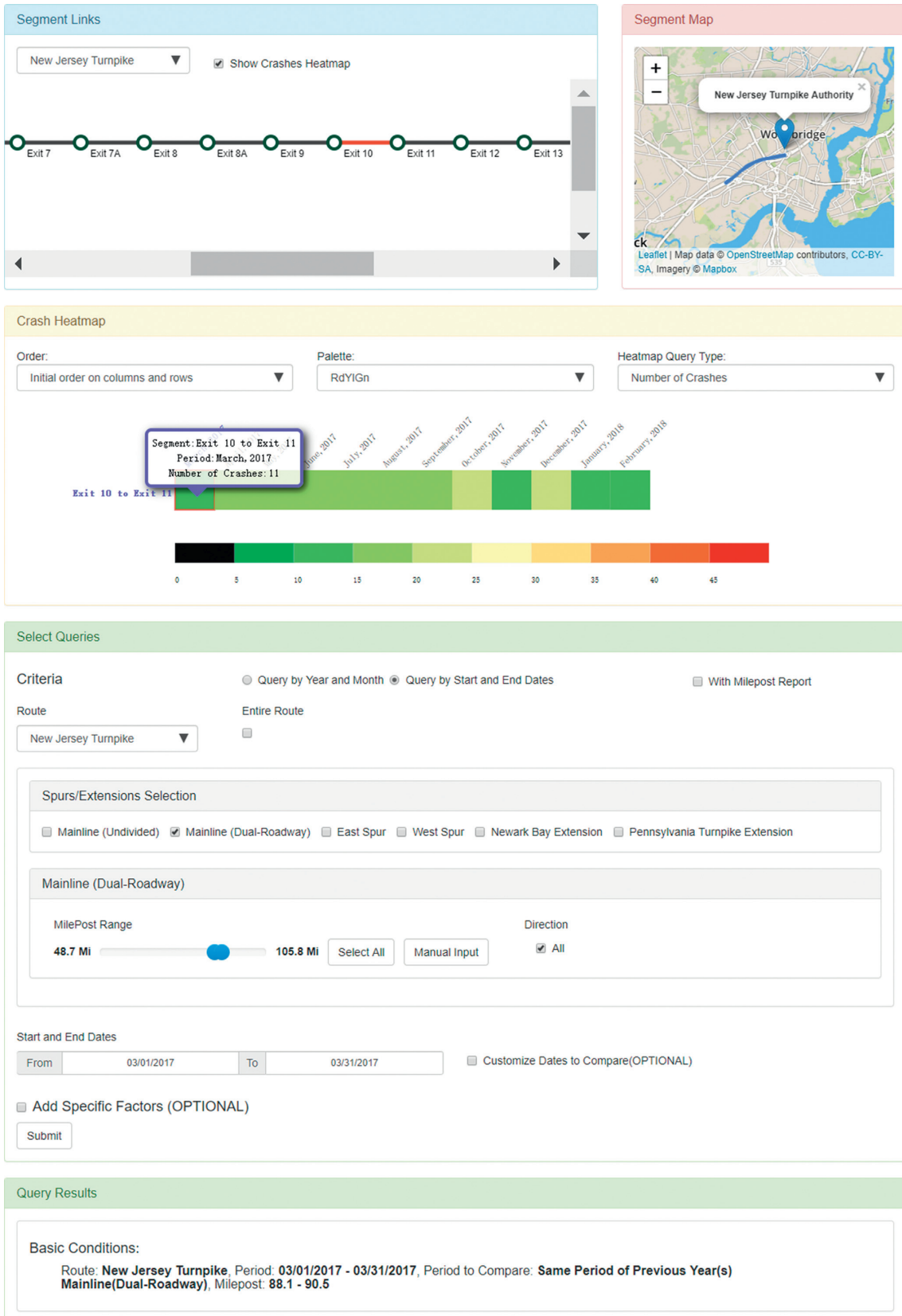


FIGURE 5: Illustration of UI of segment map and heatmap query.

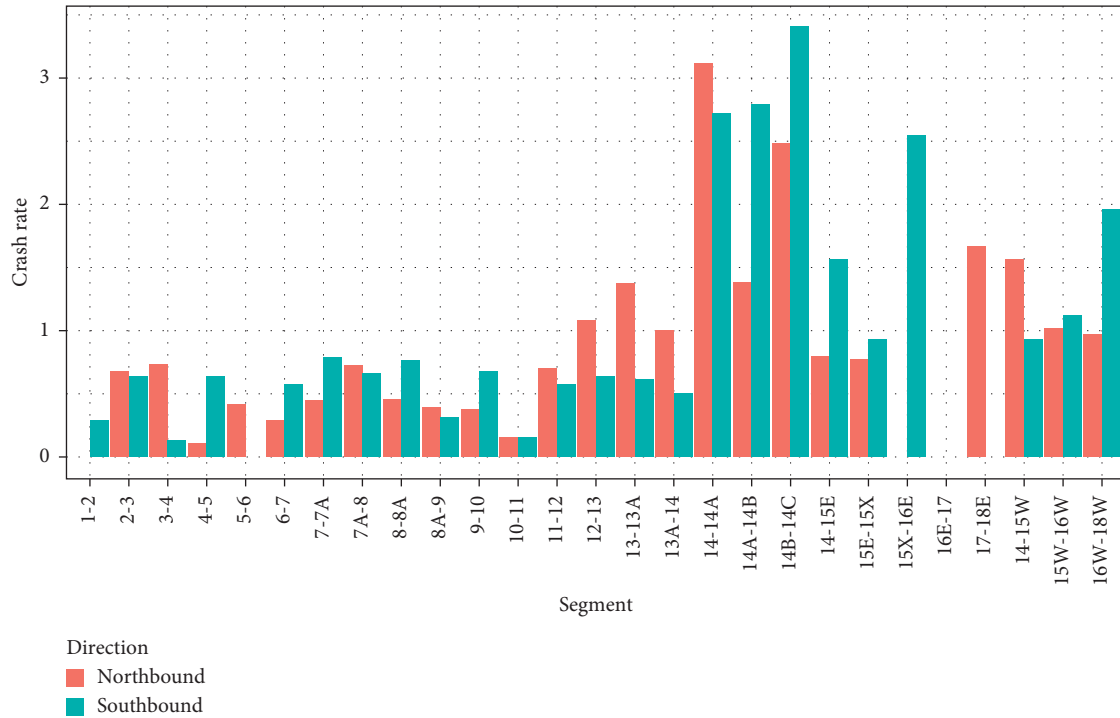


FIGURE 6: Crash rates of NJTP segments in March 2017.

TABLE 2: Results of speed tests.

Query type	Roadway	Date	Mainline, spurs, extension, and milepost range	Conditional query	Average execution time from database (sec) ¹	Average execution time from buffer (sec) ¹
General query	All	2017 annual	N/A ²	N/A ²	9.979	0.365
	GSP	2017 annual	N/A	N/A	6.221	0.348
	GSP	March 2017	N/A	N/A	2.022	0.345
Detailed query	NJTP	3/1/2017 to 3/31/2017	All	None (all selected)	5.828	N/A
	NJTP	3/1/2017 to 3/31/2017	Mainline 0–22, mainline 79–105.8, all directions	None (all selected)	3.277	N/A
	NJTP	3/1/2017 to 3/31/2017	Mainline 0–22 (SB), mainline 79–105.8 (NB inner)	None (all selected)	1.887	N/A
	NJTP	3/1/2017 to 3/31/2017	All	Day(s) of week: Sunday, Monday, Tuesday, apparent contributing factor(s): 01 unsafe speed, 04 failed to yield, 08 improper turning, location type: main roadway, service area	1.639	N/A
	NJTP	3/1/2017 to 3/31/2017	All	Road condition: 01 dry, 02 wet, 04 icy	6.335	N/A
With MP level query	NJTP	3/1/2017 to 3/31/2017	All	None (all selected)	6.240 ³	N/A

¹In general query, results are stored in JSON buffer and can be fetched later. JSON buffer is not available in detailed query and MP level queries. Execution times include data processing, database connection, query execution, and results visualization times. ²In general query, milepost range and conditional queries are not available. ³The time includes MP level query and detailed query.

might be overcounted. To solve this issue, an additional sequence field (ACC_CASE_NUM_SEQ) was used to filter the most up-to-date record for each crash.

For the second issue, it was found that while some of the crashes did not actually occur on the study route, namely, NJTP or GSP, more careful analysis revealed the fact that they occurred on the connecting roads which are also under the jurisdiction of the agency. In those cases, the values of the route number were not specified, but this issue still had to be addressed for obtaining accurate results. Hence, a more agile location filtering function was developed which parses the Acc_case field rather than only using the location value. These issues were solved, and query results were verified to be consistent with the NJDOT crash data.

6. Conclusions and Future Work

This paper proposed an online safety analytics and visualization tool called SAVE-T for analysis of vehicle crashes which can be adopted by highway agencies who have to deal with large crash databases. The objective of the tool is to provide a convenient way for automatically generating key analytics from the crash datasets and consequently to provide better insights to safety experts and decision-makers. One of the main features of the tool, general query, is based on the NJDOT format for crash reporting purposes and SAVE-T automates this labor-intensive task. The tool also provides convenient options for visualization, exporting, and validation (query export) of the analysis results.

To further improve the query efficiency of the tool, a possible improvement can be the adoption of emerging big data technologies such as MongoDB instead of using traditional database management platforms such as Oracle and MySQL. Currently, the tool optimizes the experience of its users accessing it through their desktop web browsers. However, it can be beneficial to develop a more responsive interface which can improve the experience of smartphone users. This can be especially important for the users on-the-go such as state troopers. Future improvement will also include the development of statistical models to identify high-risk locations and the correlations between crashes and possible causes such as traffic violations and to predict crash risk in the near future.

Data Availability

The data used to support the findings of this study have not been made available because of the restriction by New Jersey Turnpike Authority.

Disclosure

The opinions and conclusions presented in this paper are the responsibility of the authors and do not reflect the views of sponsors and other participating agencies.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Study conception and design were contributed by Y. Zhu, S. Demiroglu, K. Xie, and K. Ozbay; data collection was done by Y. Zhu, S. Demiroglu, and Di Sha; analysis and interpretation of results were carried out by Y. Zhu; data visualization was done by Y. Zhu and H. Yang; draft manuscript preparation was done by Y. Zhu, S. Demiroglu, and K. Ozbay. All authors reviewed the results and approved the final version of the manuscript.

Acknowledgments

This work was partially funded by New Jersey Turnpike Authority (NJTA), and the grant was administered by Rutgers University and New York University. The authors thank NJTA for providing the data, guidance, and suggestions at various stages of this study. This study was also partially supported by the Young Scientists Fund of the National Natural Science Foundation of China (Grant no. 61903205) and Young Scientists Fund of Natural Science of Inner Mongolia (Grant no. 019BS07002).

References

- [1] World Health Organization, "Top 10 causes of death," 2016, http://www.who.int/gho/mortality_burden_disease/causes_death/top_10/en/.
- [2] NHTSA, "USDOT releases 2016 fatal traffic crash data," 2017, <https://www.nhtsa.gov/press-releases/usdot-releases-2016-fatal-traffic-crash-data>.
- [3] Fortune, "2016 was the deadliest year on American roads in nearly a decade," 2017, <http://fortune.com/2017/02/15/traffic-deadliest-year/>.
- [4] New Jersey Motor Vehicle Commission, "New Jersey NJTR-1 crash report manual," 2017, <https://www.state.nj.us/transportation/refdata/accident/pdf/NJTR-1CrashReportManual12517.pdf>.
- [5] R. Meyer, "After 61 years, America's busiest highway is almost complete," 2018, <https://www.theatlantic.com/technology/archive/2018/01/after-61-years-americas-busiest-highway-is-almost-complete/550982/>.
- [6] New Jersey Department of Transportation, "Crash data," 2017, https://www.state.nj.us/transportation/refdata/accident/crash_data.shtm.
- [7] R. A. Krammes, K. S. Rao, and H. Oh, "Highway geometric design consistency evaluation software," *Transportation Research Record*, no. 1500, pp. 19–24, 1995.
- [8] D. W. Harwood, D. J. Torbic, K. R. Richard, and M. M. Meyer, *SafetyAnalyst: Software Tools for Safety Management of Specific Highway Sites*, Turner-Fairbank Highway Research Center, McLean, Virginia, 2010.
- [9] University of California Berkeley, "Transportation injury mapping system (TIMS)," <https://tims.berkeley.edu/>.
- [10] New Jersey Department of Transportation, "State of New Jersey police crash investigation, report no. NJTR-1," 2017, <http://www.state.nj.us/transportation/refdata/accident/pdf/NJTR-1.pdf>.
- [11] American National Standards Institute, *Manual on Classification of Motor Vehicle Traffic Accidents*, American National Safety Council Illinois, IL, USA, 2007.