Automated Statewide Estimation of Crash-Induced Delay and Queueing using Crowdsourced Data

Abolfazl Karimpour (Corresponding Author)
Assistant Professor
College of Engineering
State University of New York Polytechnic Institute
100 Seymour Rd, Utica, NY 13502
Email: karimpa@sunypoly.edu
ORCiD : 0000-0002-8707-6408

Anthony Altieri
Undergraduate Researcher
College of Engineering
State University of New York Polytechnic Institute
100 Seymour Rd, Utica, NY 13502
Email: altiera@sunypoly.edu

Adrian Cottam
Assistant Research Professor
Auburn University Transportation Research Institute
Auburn University
Room 205, 311 W Magnolia Ave, Auburn, AL 36849
Email: adrian.cottam@auburn.edu
ORCiD : 0000-0001-5654-4347

Ellwood Hanrahan II
Statewide Mobility Services Program Manager
NYS Department of Transportation
50 Wolf Road Albany, NY 12232
Email: ellwood.hanrahan@dot.ny.gov

Word Count: 6,955 + 2 * tables (250) = 7,455 words

Submitted on

ABSTRACT

Due to the unpredictable nature of crashes, accurately predicting when crashes will happen is challenging. Therefore, a key strategy for enhancing safety focuses on mitigating the impact of crashes when they do occur. Many agencies have adopted this approach by implementing incident management programs designed to reduce congestion and prevent secondary crashes. These programs require quick and efficient responses, which depend on timely and relevant information, such as accurate estimates of crash-induced congestion. This study introduces a method for estimating crash-induced delay and traffic congestion queue length, using machine learning and fusing multiple data sources. Police-reported crash data and Waze crowdsourced data were collected for all thruways in New York State. A Long Short-Term Memory model was trained utilizing the spatiotemporal alignment of these data sources. The model provides statewide estimations across various crash types and severity levels while considering roadway, surface, and weather conditions. The model demonstrated a root mean squared error of 0.85 minutes for estimating crash-induced delays and 1.41 miles for queue length estimation. Additionally, the performance of the proposed model was compared with three conventional models: DummyRegressor, Random Forest, and XGBoost. The results showed that our model outperformed the other three models while estimating crash-induced delay and queue length. The findings of this study can be applied to roadway planning and driver navigation by displaying accurate crash information on variable message signs. This information could help drivers make informed route choices, including potential detours, while also providing valuable data to roadway agencies to prevent secondary crashes.

Keywords:

Crash-Induced Delay, Long Short-Term Memory Model, Crash-Induced Traffic Congestion,
Variable Message Signs, Assisted Route Choice

INTRODUCTION

Motor vehicle traffic crashes present ongoing concerns for transportation professionals, affecting both safety and mobility. In the United States, motor vehicle crashes are alarmingly frequent, with approximately 5.93 million police-reported crashes in 2022 alone, resulting in about 2.38 million injuries and 42,514 fatalities (1). These crashes not only pose significant safety risks but also contribute to traffic congestion and delays (2, 3). Roadway congestion can create a self-perpetuating cycle where an initial crash causes traffic delays, which in turn leads to secondary crashes (4). Due to the unpredictable nature of crashes, one practical approach to enhancing safety is to mitigate their impacts once they occur. Many agencies have developed incident management programs aimed at reducing congestion and, consequently, secondary crashes (5, 6). Effective traffic incident management requires rapid responses and access to accurate information about current roadway conditions, including estimates of potential delays and queue lengths that might form. It is equally important to inform travelers about roadway conditions and expected delays, allowing them to choose alternative routes to avoid crash sites. Obtaining real-time roadway information at crash locations can be challenging due to the distributed nature of crashes and the lack of nearby roadway sensors. To address this, a promising approach involves estimating traffic conditions, including anticipated delays, for crashes using advanced data analysis methods. This can enhance both the response to incidents and the dissemination of critical information to travelers, ultimately improving safety and mobility on the roads.

Several studies have focused on estimating the spatial or temporal effects of crashes, as well as the societal costs associated with crashes (2, 7-11). In terms of congestion caused by crashes, the spatial effects are typically considered to consist of traffic queueing, while the temporal effects are typically considered to consist of traffic delays. Crash-related congestion, also known as incident-induced non-recurrent congestion, must be quantified before it can be estimated. Therefore, several studies have focused on identifying incident-induced non-recurrent congestion (12-15). Several methods have been applied to estimate different target metrics of spatiotemporal effects of crash-related congestion, using several different data sources, as summarized in

Table 1.

Table 1 Summary of Crash-Induced Delay Estimation Studies

Study	Method(s)	Target Metric(s)	Data Source
Wang, Hallenbeck et al. 2008 (11)	Deterministic queuing theory	Delay	Loop detectors, crash data
Chung 2011 (13)	Empirical methods	Delay	Vehicle detection systems, crash data
Chung and Recker 2012 (10)	Binary integer programming (BIP)	Delay, congested region (roadway segments)	Loop detectors, crash data
Li and Chen 2013 (16)	Multi-layer perceptron (MLP)	Travel time	Loop detectors, electronic toll collection (ETC) system, precipitation data, crash data
Chen, Liu et al. 2016 (17)	K-nearest neighbor (KNN)	Delay, congested region	Loop detectors, crash data

Zheng, Qi et al. 2021 (18)	Integer programming model	Delay, congested region	Loop detectors, GPS data, simulation, crash data
Lian and Loo 2024 (2)	Gradient boosting regression, GPS map matching	Delay, delay cost	GPS data, traffic volume sensors, point of interest (POI) data, crash data

As detailed in Table 1, while methods used in recent studies vary considerably, most studies evaluate crash-induced delay using sensor-based data, which is used in every study. While sensor-based data is valuable, it is not always widely available. This makes more widespread data sources, such as crowdsourced data, preferable for large-scale analyses.

Crowdsourced data has been successfully applied for large-scale estimations of traffic parameters such as traffic volumes, speeds, and travel times, both independent of traffic sensors and complementing traffic sensors (19-24). As there are complex and oftentimes non-parametric relationships between crowdsourced data and traffic parameters, many recent studies have applied non-parametric machine learning and deep learning models to estimate traffic parameters. Some of the models effectively used to estimate traffic parameters using crowdsourced data include extreme gradient boosting (XGBoost) (20, 21, 23), random forest models (25), artificial neural networks (20), and long-short-term-memory (LSTM) networks (21, 22). While some recent studies listed in Table 1 apply machine learning algorithms or crowdsourced data to estimating spatiotemporal crash effects, there are some limitations to consider.

The study conducted by Li and Chen used a multi-layer perceptron (MLP) model to estimate travel times using multi-source data (16). While this study provides an innovative application of neural networks to estimate travel time during non-recurrent congestion, it relies on loop detector data and ETC data, limiting the scale of its application to only locations where sensors are available. This can be problematic, as sensors can be prone to failure, and depending on a transportation agency's funding level or location, the number of sensors can be limited (21, 26). This limitation is also present in the study by Chen et al., where a KNN machine learning method was used to estimate delay based on loop detector data (17). A study by Zheng et al. used probe vehicle GPS data, which is similar to crowdsourced speed data, but does not have the same level of coverage as it was part of a specific study area (18). Furthermore, this study’s use of loop detector data similarly limits its applicable spatial area to only locations near sensors. Finally, in a study by Lian et al., a gradient boosting regression is used with a GPS map-matching methodology to use taxi GPS data to estimate delays and their associated costs (2). This paper used an innovative approach to estimating volumes using a gradient-boosting regression to account for the limited spatial availability of traffic volume sensors. However, the paper uses taxi GPS data rather than crowdsourced data, limiting its applicability to locations within taxi service areas. In some countries or locations, this could have relatively high coverage, but in many cases, it will not.

To address the limitation of limited spatial coverage when estimating crash-induced delay and queueing, this research introduces a novel method for estimating crash-induced delays and the spatial boundaries of crash-induced traffic congestion (queue length) using machine learning and crowdsourced data. The crowdsourced data is used to provide ground-truth crash-induced queuing and delays, and a machine learning model is trained to estimate these parameters from police crash report data. This allows for predictive estimates of the expected delay and queues to be expected from a crash as soon as it occurs. A Long Short-Term Memory (LSTM) model, known for its ability to capture temporal dependencies in sequential data, is employed to perform large-scale estimations for various crash types and severity levels. By fusing police-reported crash data with crowdsourced data collected from smartphones, our approach overcomes scalability issues that often constrain sensor-based studies. This allows for more comprehensive analysis and application across a wide geographic area. The specific contributions of this study can be summarized as follows:

- An LSTM model is trained using crowdsourced data and crash reports to predictively estimate crash-induced delay and queueing from crash report descriptors and current road conditions
- A method to spatially fuse crash reports with associated crowdsourced data is developed
- The state of New York is used as a case study to validate the proposed approach, with statewide queue length and delay estimations performed for crashes on all NY thruways over three years.

STUDY DESIGN & DATA COLLECTION

Site Selection

To develop and test the proposed estimation model, data from all thruways in New York State were collected. The Thruway network encompasses several key routes, including Interstate 87, which connects New York City to Albany; Interstate 95, linking New York City to Connecticut; Interstate 287, which joins I-87 with I-95; Interstate 90, covering both the Berkshire Spur that connects I-87 to the Massachusetts Turnpike and the mainline Thruway running from Albany to the Pennsylvania border through Syracuse and Buffalo; and Interstate 190, connecting Buffalo to Niagara Falls, with annual VMT of approximately 8 billion vehicle miles. In recent years, significant crashes have occurred on these thruways, with 17 fatal crashes and 18 fatalities in 2021, 11 fatal crashes and 14 fatalities in 2022, and 25 crashes resulting in 27 fatalities in 2023. For this study, we focused exclusively on thruways to minimize the impact of side streets and intersections, significantly enhancing the accuracy of our model. By concentrating on major routes that are less affected by local roadway variations, we aimed to ensure a more reliable assessment of traffic conditions. This strategic choice allowed us to capture traffic dynamics more accurately, yielding results that better reflect true congestion patterns. Furthermore, according to the Federal Highway Administration (FHWA), traffic incident management, which is an application of our method, is primarily focused on freeways due to the potential for significant delays and safety risks caused by incidents (27).

Figure 1 illustrates the study sites where we collected data, highlighting the thorough and focused nature of our research approach. The map data was obtained by directly querying the OpenStreetMap API, and the base map itself is from OpenStreetMap(28)

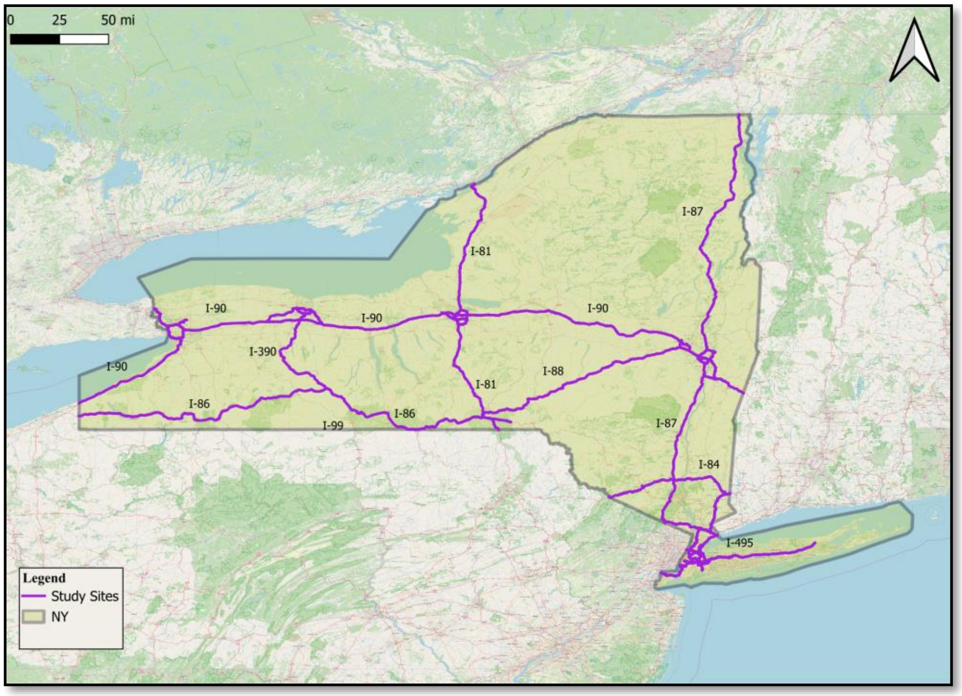


Figure 1 Study corridors: NY Thruways

Data Collection

Data for this study was gathered from two primary sources over a period of three years (2021 to 2023) from all thruways in New York State, as illustrated in Figure 1. The first source was traditional police-reported crash data, which adheres to the Model Minimum Uniform Crash Criteria (MMUCC) and includes detailed information across three main tables: the crash table (documenting crash date, time, severity, weather conditions, surface conditions, and road type), the vehicle table (providing details on the number of vehicles involved, vehicle classifications, and related specifics), and the person table (offering data on demographics such as gender, age, and other geographical details). In addition to police-reported crash data, crowdsourced Waze data was collected over the study period. Waze, a navigation app developed by Google, uses user-reported incidents and historical data to inform users about travel disruptions, including traffic jams, accidents, hazards, construction, and road closures. Waze data comprises general information, such as timestamps and geographic areas, traffic alerts reported by users, and traffic jam information generated based on a user's location and speed. Access to the Waze Data Feed, which powers these alerts, was provided through the New York State Department of Transportation (NYSDOT).

Figure 2 illustrates the spatial distribution of the two data sources over the study period. This figure is color and shape-coded to differentiate between police-reported crashes and crowdsourced Waze data. As shown in Figure 2, there is a significant overlap between the two data sets. However, due to the nature of crowdsourced data, which relies on user input, some duplication exists within the Waze data. This overlap and the potential for redundancy highlight the importance of integrating and cross-referencing multiple data sources to ensure accuracy and comprehensiveness in analyzing traffic patterns and incident impacts.

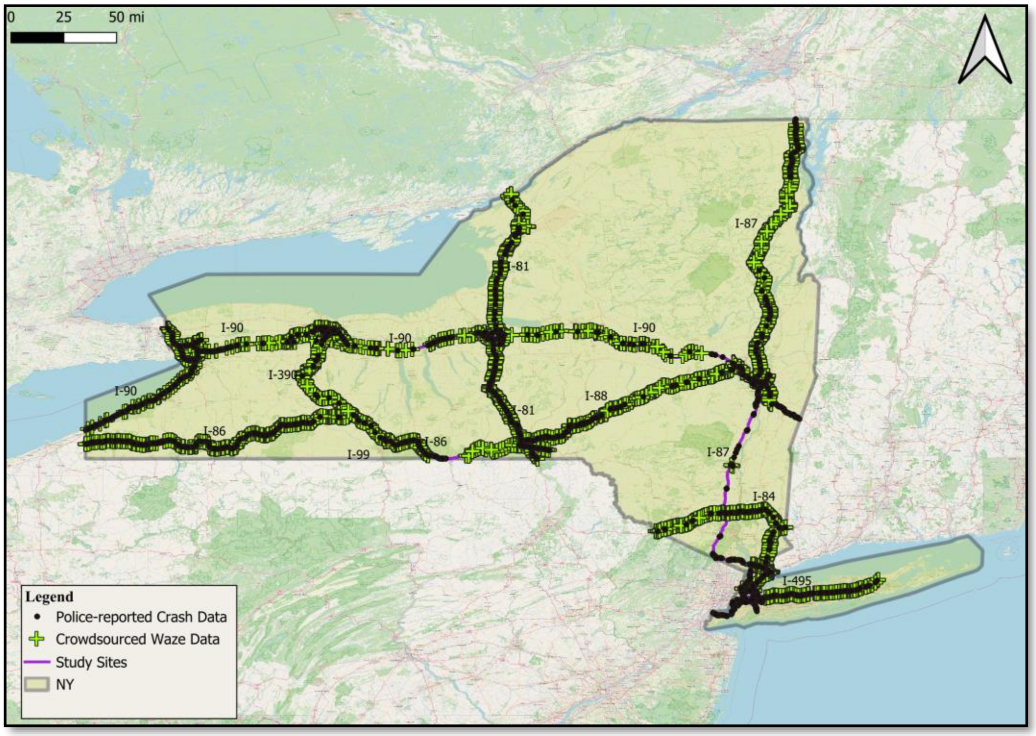


Figure 2 Spatial data distribution (police-reported crash and crowdsourced Waze data)

The datasets from both police-reported crash data and crowdsourced Waze data contain numerous columns with a wide range of information, many of which overlap. Table 2 presents the list of data variables that were processed and prepared for modeling. These variables were collected from both data sources to ensure a comprehensive and insightful analysis. By focusing on the most relevant and informative variables, the study aimed to enhance the accuracy and effectiveness of the predictive model, ultimately estimating crash-induced delay and queue length. The variables with a subscript of "O" represent the model output, those with "I" represent the model input, and those with "F" represent the variables used for filtering.

Table 2 Data Variables

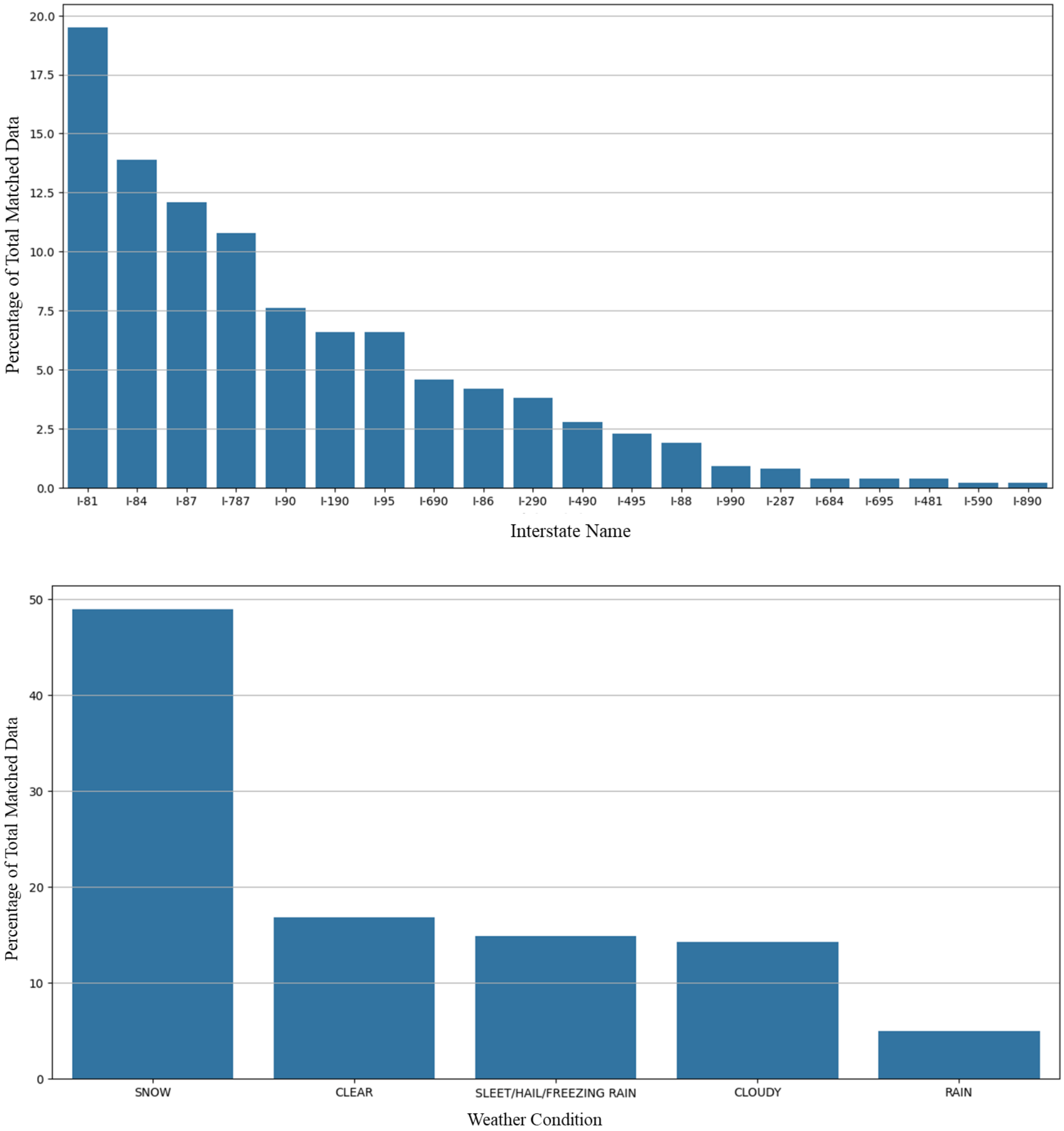
	Variables	Description	Variable Type
Crowdsourced Waze Data	Crash Delay (O)	Amount of delay per minute	Numerical
	Crash Queue Length (O)	Length of queue generated due to the crash	Numerical
	Average Speed (I)	Speed of user when reporting the crash	Numerical
	Congestion Level (I)	Level 0: Free flow Level 1: Low Level 2: Medium Level 3: High Level 4: Very high Level 5: Blocked traffic	Categorical
Police-reported Crash Data	Location (I)	The longitude and latitude location for the start of the queue	Numerical
	Time (I)	The time when the crash was reported by the user	Numerical
	Month (I)	Month of Year	Numerical
	Year (I)	Year	Numerical
	Weekday/Weekend (I)	Day of week	Numerical
	Vehicle Type (I)	Passenger vs. commercial vehicle	Categorical
	County (F)	The county where the crash happened	Text String
	Weather Condition (I)	Weather type when the crash happened (cloudy, rainy, snow, hail, and clear)	Categorical
	Light Condition (I)	Light conditions at the time of the crash (dark, dusk, daylight, and dawn)	Categorical
	Surface Condition (I)	Surface condition at the time/location of crash (wet, dry, flooded water, snow, and slush)	Categorical
	Posted Speed (I)	The speed limit on the roadway	Numerical
	Crash Characteristics (I)	The type of crash, and crash apparent reasons	Text String

To incorporate the diverse data types presented in Table 2 into our model, each type requires specific processing, such as encoding or tokenization. The following section describes the preprocessing steps applied to the data to prepare it for use in the model.

Data Matching and Preprocessing

Since Waze data is a user-input source, it may contain duplications and inaccuracies. To address these issues, the initial step involved validating and spatiotemporally merging Waze data with police-reported crash data to eliminate duplication and incorrect inputs. Combining both datasets provided a robust feature set for the estimation model. To identify matches between the two datasets, a spatial and temporal thresholding approach was employed. This method involved several steps: Initially, a cartesian product of Waze data and crashes occurring on the same day was constructed, generating all possible combinations irrespective of specific locations or times. Next, the time difference between each crash and Waze data pair was calculated, with pairs having a time difference exceeding thirty minutes being discarded. Subsequently, the straight-line distance between the crash and Waze data was computed by converting latitude and longitude coordinates into (x, y) coordinates in miles. Pairs with a distance greater than two miles were excluded, leaving only those within thirty minutes and two miles of each other. This approach facilitated the matching of multiple Waze reports to a single crash, as different users might report the same traffic jam. Finally, any duplicate data entries were removed.

Figure 3 depicts the percentage of the matched data distribution across various study interstates, weather conditions, and times of day.



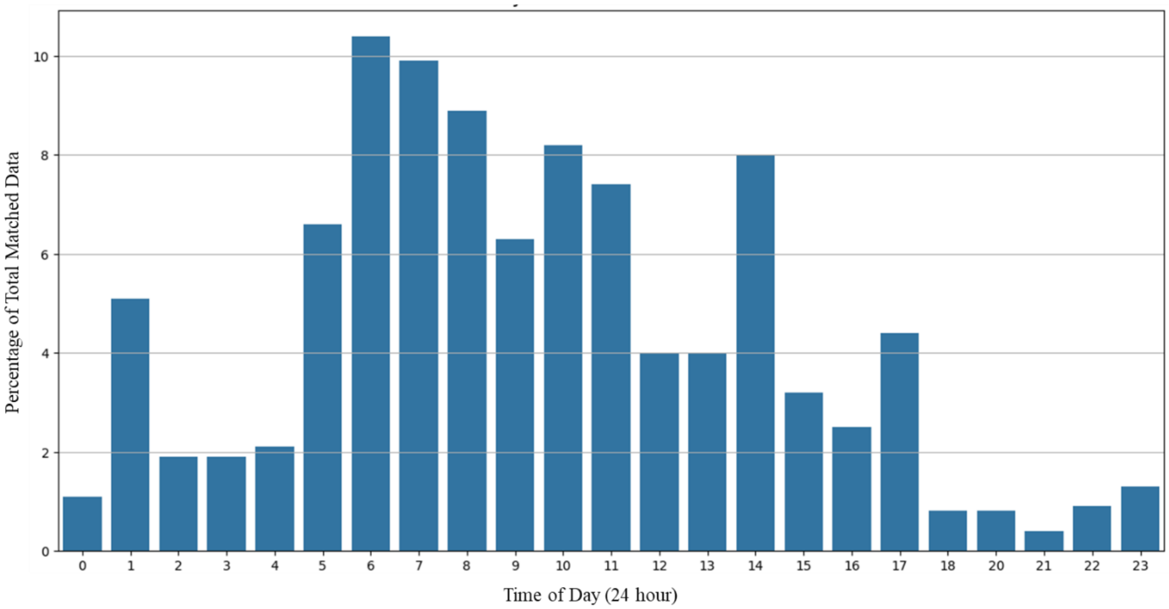


Figure 3 Percentages of matches between police-reported data and crowdsourced waze data

Figure 3-a depicts the matched data distribution percentage across various study interstates. From this figure, it can be observed that most match instances occur on Interstate 81 (I-81), a major north-south highway that serves as a vital corridor connecting upstate New York to the Great Smoky Mountains of Tennessee. This high percentage of matches on I-81 suggests that this interstate experiences significant traffic crashes and congestion, making it a critical area for monitoring and traffic management interventions. Figure 3-b illustrates the weather conditions during which most matched instances occurred, with a notable concentration during snowy conditions. This indicates that adverse weather significantly impacts traffic flow and crash occurrence, highlighting the importance of targeted strategies to manage congestion during winter months. Figure 3-c demonstrates that most matches between police-reported crash data and crowdsourced Waze data happen during daylight hours. This suggests that traffic crashes are more frequent during daytime, possibly due to increased traffic volumes during daytime hours. Understanding this trend can assist in optimizing resource allocation for traffic management and emergency response teams, ensuring they are more active during peak times.

To prepare the numerical and categorical data for modeling, we utilized three encoding techniques: cyclical, ordinal, and one-hot encoding. It is important to note that text string data will be processed through tokenization during the modeling phase. Cyclical encoding was specifically applied to temporal data, such as time of day, day, and month, to maintain the data's inherent cyclical patterns. Cyclic encoding addressed the issue of discontinuity in cyclic data, enhancing the accuracy of predictive models by capturing temporal dependencies and cyclical patterns. This approach proved particularly useful for time series data, aiding the model in understanding trends and making accurate forecasts. By treating the start and end of a cycle as connected, cyclic encoding ensured that the inherent structure of the data was preserved, resulting in a more robust and insightful analysis. For instance, in representing months of the year as 1-12, January (1) and February (2) were closely related, while December (12) and

January (1) were also close despite their numeric labels not reflecting this. The general formulas for cyclic encoding applied in this study were:

$$\text{sin}\_encoded = \sin\left(\frac{2\pi x}{T}\right)$$
$$\text{Cos}\_encoded = \sin\left(\frac{2\pi x}{T}\right)$$

where T represents the period of the cycle, and *x* is the variable we seek to encode.

Ordinal encoding was subsequently applied to columns where categorical data exhibited a specific order, such as crash severity, light conditions, and maximum injury. For example, the crash severity column utilized the KABCO scale. Ordinal encoding involves assigning numeric values to each category, thereby preserving the inherent ordered relationships among them. Lastly, one-hot encoding was used for categorical data without an inherent order, such as road surface condition, weather condition, roadway characteristics, and traffic division. For example, the weather condition column, which originally contained categories like cloudy, clear, unknown, rain, sleet/hail/freezing rain, and snow, was transformed so each category became its own column. A “1” was placed under the column of the category that appeared in the original data, allowing these categories to be numerically represented without creating artificial biases.

METHODOLOGY

Model Development Approach

To develop an accurate estimation model for crash-induced delays and queue length, a Long Short-Term Memory (LSTM) model was employed. The LSTM model is a complex type of recurrent neural network (RNN) architecture, specifically designed to effectively capture long-term dependencies in sequence prediction tasks (29). Its unique ability to retain and utilize past information makes it an exceptional choice for tasks involving temporal sequences. Unlike traditional RNNs, which often struggle with vanishing gradient problems that limit their ability to model long-range dependencies, LSTMs feature a gating mechanism that regulates the flow of information through the network. This feature is particularly beneficial for time series prediction (30) and other domains where the sequence and context of input data are crucial for accurate modeling (31).

The LSTM architecture was selected due to its exceptional ability to utilize historical data for making accurate predictions about current and future events. This capability is essential when estimating crash-induced delays, as past traffic patterns and conditions offer valuable insights into future congestion scenarios. Furthermore, LSTMs are distinguished for their robustness against noise and irrelevant data, enabling the model to focus on meaningful patterns while filtering out distractions effectively (32). The advanced architecture of LSTMs not only enhances prediction accuracy but also aids in developing more effective traffic management strategies by uncovering the underlying patterns and relationships in traffic data (33-35). Additionally, the LSTM model is adept at handling text string data and correlations among independent variables, especially those present in temporal data, further strengthening its predictive capabilities (36).

A separate model was trained for each parameter of interest, including crash-induced delay and queue length. Each model utilized the variables listed in Table 2 as independent variables. It is important to note that the numerical variables (e.g., speed limit) were included directly, while the categorical variables (e.g., road type, crash severity, light condition) were encoded as discussed in the data processing section. The text string variables (e.g., crash characteristics) were tokenized by mapping words in the dataset to unique numeric values, transforming input phrases and descriptions into zero-padded sequences of numbers. These sequences were then fed into embedding layers, which output multi-dimensional vectors. The vectors were concatenated together and input into an LSTM layer.

LSTM network consists of a series of repeating modules, or cells, each containing three primary gates: a Forget Gate that determines which information to discard from the cell state, an Input Gate that decides what new information to store in the cell state, and an Output Gate that decides what information to output from the cell state. In summary, the LSTM model used in this study effectively processes sequences of data over time, capturing the evolution of crash, roadway, and weather-related features. The model receives the independent variables as data input, with its gates and state updates allowing it to retain relevant past information and discard irrelevant information, focusing on the factors that genuinely impact traffic delays. The final hidden state h_t captures all learned information about the sequence, and the fully connected layer translates this into a specific estimation—in our case, crash-induced delays and the spatial boundaries of crash-related congestion. The customized proposed LSTM model consists of the following steps (37-39):

1. Forget Gate

The forget gate plays a crucial role in determining which historical features are less relevant to the current prediction. This process is mathematically represented by Equation 1:

$$f_t = \sigma\left(W_f \times [h_{t-1}, x_t] + b_f\right) \qquad \text{Eq. 1}$$

where f_t is the forget gate vector, which indicates the proportion of the previous state to retain, W_f is the weight matrix for the forget gate, used to weigh the input features and the previous hidden state, h_{t-1} is the output from the previous LSTM cell, representing the historical context, x_t is the vector of input features at time *t*, including independent variables such as road type, crash severity, interstate name/number, weather condition, light condition, surface condition, posted speed, etc., and b_f is the bias term for the forget gate, added to adjust the output of the gate.

2. Input Gate

The input gate regulates what new information to add to the cell state, essentially deciding how much the current crash/roadway/weather-related features contribute to the estimation model. The input layer gate has two outputs i_t and $\tilde{C}_t$, demonstrated in Equations 2 and 3.

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \qquad \text{Eq. 2}$$

$$\tilde{C}_t = tanh(W_C \times [h_{t-1}, x_t] + b_C) \qquad \text{Eq. 3}$$

where i_t is the input gate vector, indicating the proportion of the new information to be added to the cell state, $\tilde{C}_t$ represents the candidate values for the cell state, determined by applying the hyperbolic tangent function ($tanh$), σ is the sigmoid activation function, which scales the input between 0 and 1, W_i and W_c are weight matrices for the input gate and the candidate cell state, respectively. They define how much each input feature and the previous hidden state contribute to the gate's operation, and b_i and b_c are bias terms for the input gate and the candidate cell state, used to adjust the outputs. The outputs from the input gate, along with those from the forgot gate, are used to update the current cell state C_t, allowing the LSTM to selectively integrate new information while retaining valuable historical information.

3. Updating the Cell State

In this step, the outputs from the input and forget gates are used to update the current cell state C_t as shown in Equation 4:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \qquad \text{Eq. 4}$$

where f_t is the forget gate vector, which decides how much of the previous cell state should be retained, i_t is the input gate vector, determining the amount of new information to be added, $\tilde{C}_t$ represents the candidate values for the new cell state, reflecting the processed new information, and C_{t-1} is the previous cell state, containing the historical context. In this equation, the previous cell state $C_{t-1}$is scaled by the forget gate f_t to decide how much past information should be retained. Simultaneously, the input gate i_t scales the candidate cell state $\tilde{C}_t$ to add the new information, thus updating the cell state $\tilde{C}_t$ with a combination of past and present information.

4- Output Gate

The output gate plays a crucial role in determining the extent to which information from the present cell state is outputted from the LSTM cell. This process is governed by Equations 5 and 6:

$$O_t = \sigma(W_O \times [h_{t-1}, x_t] + b_O) \qquad \text{Eq. 5}$$

$$h_t = O_t \times tanh(C_t) \qquad \text{Eq. 6}$$

Equation 6 illustrates how the hidden state h_t is derived from the current, updated cell state C_t and the output gate value O_t. The hyperbolic tangent function ($tanh$) ensures that the cell state's information is properly scaled. The hidden state at the final time step h_t is typically passed through a fully connected layer to estimate crash-induced delay or queue length, as shown in Equation 7:

$$y_t = W_y . h_t + b_y \qquad \text{Eq. 7}$$

By employing the LSTM architecture tailored for this case study, the model leverages the temporal relationships and interactions between crash types, severity levels, and differing roadway, surface, and weather conditions variables in Table 2, enabling accurate and insightful estimations.

Model Evaluation Metrics

The performance of the proposed LSTM model was evaluated using several conventional metrics:

1) Mean absolute error (MAE): This measure shows the average imputation error (the average distance of the imputed value from the actual value).

$$MAE = \frac{\sum_{i=1}^{n} |y_t - \widehat{y_t}|}{n} \qquad \text{Eq. 8}$$

where *n* is the number of sample data, y_t is the observed crash-induced delays (or crash-induced traffic queue length), and $\widehat{y_t}$ is the estimated value. Previous studies such as (40-42) suggested that MAE is a great approach to evaluate the effectiveness of the estimation.

2) Mean squared error (MSE): This measure shows the average squared difference between the predicted and actual target values in a dataset.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_t - \widehat{y_t})^2 \qquad \text{Eq. 9}$$

(10)

3) Root-mean-square error (RMSE): This measure is the weighted average of imputation error that gives higher weights to the larger errors; RMSE is useful when the magnitude of the error is important. RMSE can be calculated using equation 10.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_t - \widehat{y_t})^2} \qquad \text{Eq. 10}$$

4) Mean Absolute Percentage Error (MAPE): This metric indicates the average difference between the estimated and actual values, expressed as a percentage of the actual values. MAPE is a good measure of error because it provides a clear and interpretable indication of accuracy by normalizing errors, allowing for easy comparison across different scales and datasets.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n} \frac{|y_t - \widehat{y_t}|}{y_t} \qquad \text{Eq. 10}$$

RESULTS

Model Evaluation

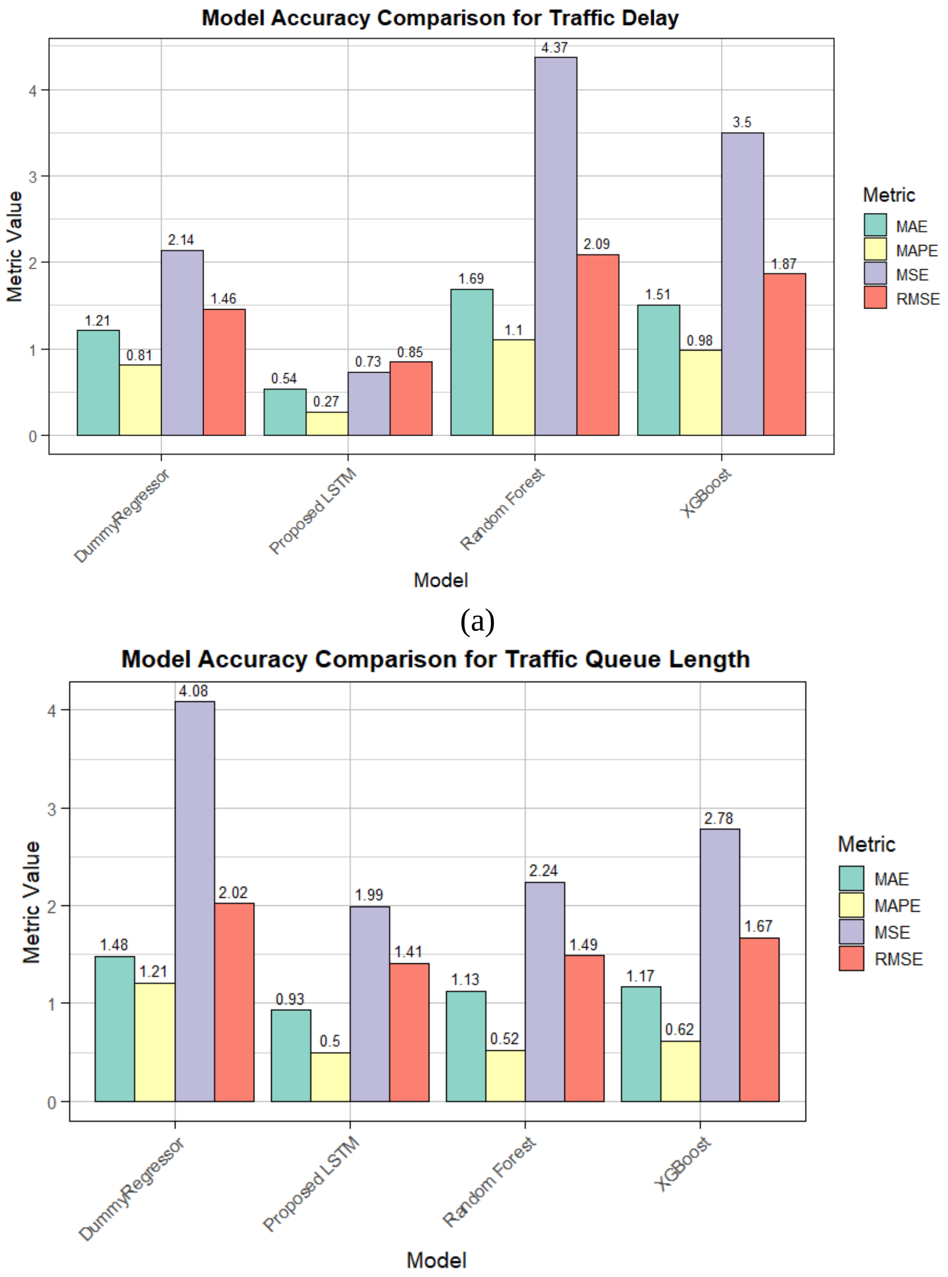
Separate models were trained for each parameter of interest, including crash-induced delays and traffic queues. Each model was designed using the variables listed in Table 2 (e.g., road type, crash severity, light condition) as independent variables. The developed estimation model was employed to evaluate crash-induced delays and traffic queues using data collected from New York State's thruways.

The dataset was divided into training and testing subsets, with 80% allocated for training and 20% for testing. To optimize the model's architecture and hyperparameters, the Keras Hyperband Tuner was utilized. This tool enabled the exploration of various layer sizes, optimizers, and learning rates, systematically testing combinations to refine the model for optimal performance. The model was trained with an 80:20 train-validation split, employing the Adam optimizer with a learning rate of 0.001. The training process spanned 30 epochs with a batch size of 32, and mean squared error was used as the primary loss function.

In addition, the accuracy of the proposed LSTM model was compared with three other conventional imputation models: Dummy Regressor, Random Forest, and XGBoost. The Dummy Regressor serves as a simple baseline to compare against other more complex regressors, providing a benchmark for model evaluation. Random Forest, known for its ensemble learning approach, is used to assess the model's performance against a technique that handles high-dimensional data effectively. Finally, XGBoost is chosen for its ability to handle large datasets and capture complex patterns, offering insights into the model's performance against a gradient-boosting framework. Figure 4 illustrates the model evaluation and comparison results.

Table 4-a provides detailed results of this performance evaluation for estimating delay and Figure 4-b details results for traffic queue length estimation. Based on the results presented in Figure 4, the developed model demonstrated a high level of accuracy in estimating crash-induced delays and queue lengths using various crash and environmental-related variables. The MSE values were 0.73 for delay estimation and 1.99 for traffic queue length estimation, indicating that the model achieves relatively low error margins. The RMSE, which is often considered more interpretable due to its units, further supports these findings, with values of 0.85 minutes and 1.41 miles, respectively. The MAE shows average errors of 0.54 minutes and 0.93 miles. This suggests that the model is reasonably effective in making close approximations to the true values, though some variations still exist. Notably, the MAPE reveals a significant discrepancy between delay and traffic queue length estimations, with MAPE values of 27% and 50%, respectively. According to the literature review, this level of accuracy falls within the "good accuracy" subgroup, suggesting a more consistent and reliable performance in estimating queue lengths.

When evaluating the model's performance in terms of overestimation and underestimation, it was revealed that for delay estimation, the model underestimated 61% of the time and overestimated 39% of the time. For traffic queue length estimation, these figures were 45% underestimation and 55% overestimation. Given these results, a conservative approach may be warranted. Implementing a buffer in the estimation process could help account for potential underestimations, ensuring that safety measures and traffic management strategies are adequately prepared for real-world scenarios.



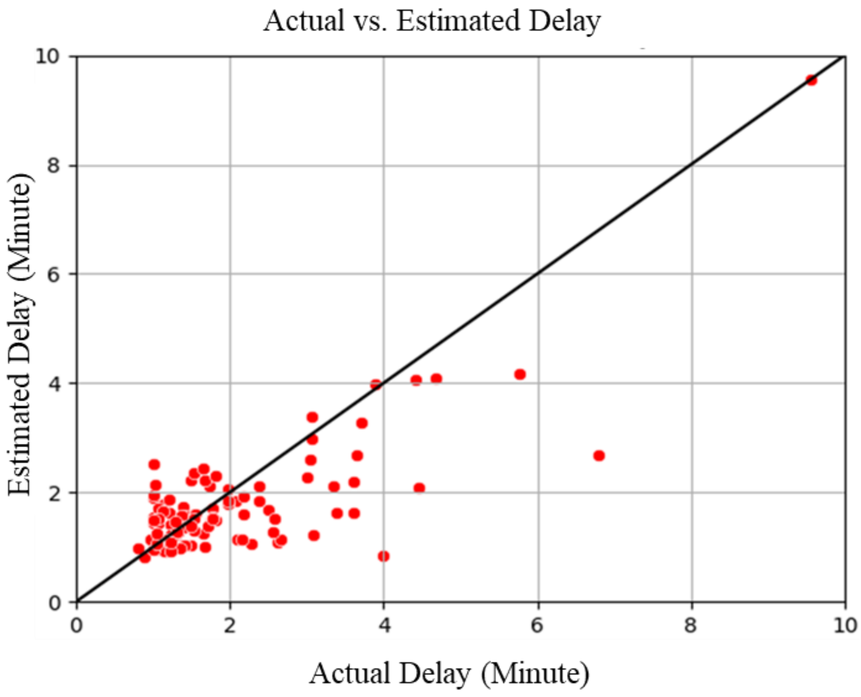
(a)

(b)

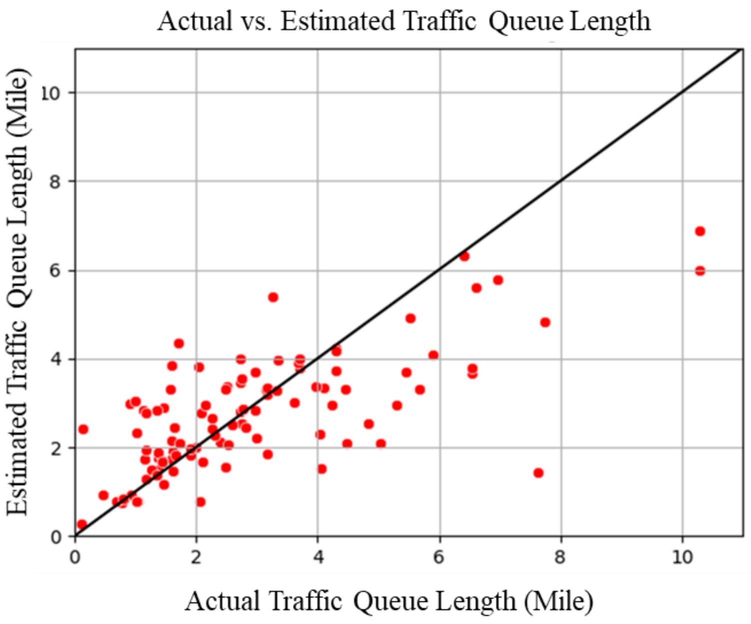
Figure 4 Model comparison results

The comparison results between the proposed model and other conventional models demonstrate that our proposed LSTM model outperformed the latter in estimating both crash-induced delay and queue length. This superior performance is attributed to the LSTM architecture's exceptional ability to leverage historical data for making accurate estimation about current and future events. Moreover, LSTMs are known for their robustness against noise and irrelevant data, allowing the model to concentrate on significant patterns while effectively filtering out distractions. Additionally, the LSTM model excels at handling text string data and managing correlations among independent variables, especially those found in temporal data, which further enhances its predictive capabilities. In this particular case study, our crash characteristics data included text strings, and we observed a high correlation among variables, which the LSTM model utilized more effectively compared with other conventional models.

To analyze the proposed model performance in different scenarios, the actual versus estimated plots are examined. Figure 5-a presents a comparison of actual versus estimated delays, while Figure 4-b displays this comparison for spatial boundaries of crashes, specifically queue lengths caused by these crashes. As shown in Figure 4-a, the model demonstrates superior performance when estimating delays of less than 4 minutes. However, the accuracy diminishes as delays increase. In Figure 5-b, it is evident that the model provides higher accuracy for estimating queue lengths of less than 4.5 miles. These observations highlight the model's effectiveness in specific scenarios, with its performance decreasing as the magnitude of the delay or queue length increases.



(a)



(b)

Figure 5 Actual vs estimated plots

The observed higher accuracy in estimating crash-induced delays compared to traffic queue lengths can be attributed to several factors. Firstly, the delay data may inherently possess more precise inputs from users, as drivers can easily perceive and report their delays, which are directly experienced in real-time. Additionally, delay estimations can be influenced by individual driving speeds and behaviors, which are more consistent and uniform across users, whereas queue length involves numerous external variables, such as roadway characteristics, road capacities, and the presence of other traffic control measures, which can lead to greater variability and reduced estimation accuracy.

Practical Application and Policy Implication

Being able to accurately estimate crash-induced delay has significant implications for traffic management and route planning. The Thruway Authority could provide real-time updates on variable message signs, thereby enhancing drivers' ability to make informed route choices. This proactive approach would not only aid in minimizing travel disruptions but also improve overall traffic flow. Additionally, route navigation applications such as Google Maps and Apple Maps could leverage these estimates to update arrival times and delay information, offering users more reliable and timely route recommendations. This integration of predictive delay data into navigation systems and traffic management tools represents a substantial advancement in optimizing travel efficiency and safety. Furthermore, the model's ability to estimate crash spatial boundaries, or queue lengths, with an accuracy of less than 2 miles has significant implications for traffic management. This precision allows the Thruway Authority to implement targeted interventions effectively, such as rerouting traffic or deploying traffic control measures. Additionally, accurately estimating the back-of-queue is crucial for reducing rear-end collisions.

This proactive approach could not only alleviate congestion but also enhance roadway efficiency. Moreover, accurate queue length estimates could be used to improve incident response strategies and enhance the real-time information provided to drivers through variable message signs and navigation apps. These applications collectively contribute to a more fluid and responsive traffic management system, ultimately improving overall travel experiences and reducing delays.

CONCLUSION

This study presents a method to estimate crash-induced delay and queue length using machine learning and crowdsourced data. Data was collected from two sources: police-reported crash data and crowdsourced Waze data for all thruways in New York State, spanning a period of three years. A LSTM model was trained using spatiotemporally fused police-reported crash data and crowdsourced data from drivers' smartphones.

The model provides estimations at scale across various crash types, severity levels, and differing roadway, surface, and weather conditions. The model's performance in estimating crash-induced delays demonstrated a MSE of 0.73 minutes, a RMSE of 0.85 minutes, and a MAE of 0.54 minutes. The MAPE for delay estimation was 0.27, indicating that the model's predictions were relatively accurate. For the estimation of traffic queue lengths, the model achieved an MSE of 1.99 miles, an RMSE of 1.41 miles, an MAE of 0.93 mile, and a MAPE for traffic queue length estimation was 0.5.

The comparison results with three conventional models—DummyRegressor, Random Forest, and XGBoost—revealed that the LSTM model outperformed all of them in estimation due to its underlying advantages. The LSTM's ability to capture temporal dependencies, handle sequential data, and manage high correlations among variables made it particularly suited for this type of analysis.

By fusing police-reported crash data with crowdsourced data, our approach overcomes scalability issues that often constrain sensor-based studies, allowing for comprehensive analysis across a wide geographic area. The LSTM model, built using data from New York State thruways, is transferable to similar traffic patterns and conditions in other regions. However, some limitations and future directions should be noted. These findings can be directly applied to roadway planning and driver navigation by displaying crash information on variable message signs. Accurate information on VMS can assist drivers in making informed route selections and potential detours, while also providing critical input to roadway agencies to prevent secondary crashes.

Due to the nature of crowdsourced data, which relies on public reporting of incidents, issues such as duplicity and varying accuracy can arise. Future research could focus on extending the temporal and spatial scope of data collection, incorporating incidents from other locations, and increasing the period of data collection. Additionally, integrating other data sources such as road geometry and Annual Average Daily Traffic (AADT) data could significantly enhance the model's accuracy. Moreover, exploring additional machine learning and artificial intelligence models, such as deep learning and hybrid models like LSTM-CNN, could further improve prediction accuracy and robustness. In conclusion, the proposed method shows great potential in improving incident management and roadway safety. By leveraging machine learning and crowdsourced data, it provides timely and accurate information that can help reduce the impact of traffic crashes and improve overall traffic flow.

ACKNOWLEDGMENTS

The authors would like to thank the NYSDOT for graciously providing the crash data and crowdsourced data that made this study possible.

AUTHOR CONTRIBUTIONS

The authors confirm their contribution to the paper as follows: study conception and design: Abolfazl Karimpour, Anthony Altieri; data collection: Abolfazl Karimpour, Anthony Altieri analysis and interpretation of results: Abolfazl Karimpour, Anthony Altieri, Adrian Cottam draft manuscript preparation: Abolfazl Karimpour, Anthony Altieri, and Adrian Cottam; manuscript review & corrections: Abolfazl Karimpour, Anthony Altieri, Adrian Cottam, and Ellwood Hanrahan. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. NHTSA. Overview of motor vehicle traffic crashes in 2022. 2024. Report No.: DOT HS 813 560.
2. Lian T, Loo BP. Cost of travel delays caused by traffic crashes. Communications in Transportation Research. 2024;4:100124.
3. Wijnen W, Stipdonk H. Social costs of road crashes: An international analysis. Accident Analysis & Prevention. 2016;94:97-106.
4. Zhan C, Gan A, Hadi M. Identifying secondary crashes and their contributing factors. Transportation research record. 2009;2102(1):68-75.
5. Ma J, Lochrane T, Jodoin P. Traffic incident management programs and benefit-cost analysis. Institute of Transportation Engineers ITE Journal. 2016;86(5):30.
6. Guin A, Porter C, Smith B, Holmes C. Benefits analysis for incident management program integrated with intelligent transportation systems operations: case study. Transportation research record. 2007;2000(1):78-87.
7. Pulugurtha SS, Mahanthi SSB. Assessing spatial and temporal effects due to a crash on a freeway through traffic simulation. Case studies on transport policy. 2016;4(2):122-32.
8. Chung Y. Identification of critical factors for non-recurrent congestion induced by urban freeway crashes and its mitigating strategies. Sustainability. 2017;9(12):2331.
9. Al-Rukaibi F, AlKheder S, AlOtaibi N, Almutairi M. Traffic crashes cost estimation in Kuwait. International journal of crashworthiness. 2020;25(2):203-12.
10. Chung Y, Recker WW. A methodological approach for estimating temporal and spatial extent of delays caused by freeway accidents. IEEE Transactions on Intelligent Transportation Systems. 2012;13(3):1454-61.
11. Wang Y, Hallenbeck ME, Cheevarunothai P. Quantifying Incident-Induced Travel Delays on Freeways Using Traffic Sensor Data [2008-02]. Transportation Northwest (Organization); 2008.
12. Anbaroglu B, Heydecker B, Cheng T. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transportation Research Part C: Emerging Technologies. 2014;48:47-65.
13. Chung Y. Quantification of nonrecurrent congestion delay caused by freeway accidents and analysis of causal factors. Transportation research record. 2011;2229(1):8-18.
14. Anbaroğlu B, Cheng T, Heydecker B. Non-recurrent traffic congestion detection on heterogeneous urban road networks. Transportmetrica A: Transport Science. 2015;11(9):754-71.
15. Skabardonis A, Varaiya P, Petty KF. Measuring recurrent and nonrecurrent traffic congestion. Transportation Research Record. 2003;1856(1):118-24.
16. Li C-S, Chen M-C. Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks. Neural Computing and Applications. 2013;23:1611-29.

17. Chen Z, Liu XC, Zhang G. Non-recurrent congestion analysis using data-driven spatiotemporal approach for information construction. Transportation Research Part C: Emerging Technologies. 2016;71:19-31.

18. Zheng Z, Qi X, Wang Z, Ran B. Incorporating multiple congestion levels into spatiotemporal analysis for the impact of a traffic incident. Accident Analysis & Prevention. 2021;159:106255.

19. Li X, Cottam A, Wu Y-J. Transit arrival time prediction using interaction networks. IEEE Transactions on Intelligent Transportation Systems. 2023;24(4):3833-44.

20. Cottam A, Li X, Wu YJ. Machine-Learning Approach for Estimating Passenger Car Equivalent Factors using Crowdsourced Data. Transportmetrica A: Transport Science. 2024.

21. Cottam A, Li X, Ma X, Wu Y-J. Large-Scale Freeway Traffic Flow Estimation Using Crowdsourced Data: A Case Study in Arizona. Journal of Transportation Engineering, Part A: Systems. 2024;150(7):04024030.

22. Cui Z, Ke R, Pu Z, Wang Y. Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Forecasting Network-wide Traffic State with Missing Values. Transportation Research Part C: Emerging Technologies. 2020;118.

23. Zhang X, Chen M. Statewide Truck Volume Estimation Using Probe Vehicle Data and Machine Learning. Transportation Research Record: Journal of the Transportation Research Board. 2023.

24. Karimpour A, Ariannezhad A, Wu YJ. Hybrid data-driven approach for truck travel time imputation. IET Intelligent Transport Systems. 2019;13(10):1518-24.

25. Hoseinzadeh N, Gu Y, Han LD, Brakewood C, Freeze PB. Estimating Freeway Level-of-Service Using Crowdsourced Data. Informatics. 2021;8(1).

26. Wu Y-J, Zhang G, Wang Y. Volume data correction for single-channel advance loop detectors at signalized intersections. Transportation Research Record: SAGE Publications Sage CA: Los Angeles, CA; 2010. p. 128-39.

27. Einstein N, Luna J. SHRP2 Traffic Incident Management Responder Training Program Final Report 2018. Report No.: FHWA-HRT-18-038

28. contributors OW. 17 March 2017 17:26 UTC.

29. Su Y, Kuo C-CJ. On extended long short-term memory and dependent bidirectional recurrent neural network. Neurocomputing. 2019;356:151-61.

30. Johny K, Pai ML, Adarsh S. A multivariate EMD-LSTM model aided with Time Dependent Intrinsic Cross-Correlation for monthly rainfall prediction. Applied Soft Computing. 2022;123:108941.

31. Louis F. Long Short-Term Memory (LSTM) Networks. 2024.

32. Qiao M, Yan S, Tang X, Xu C. Deep convolutional and LSTM recurrent neural networks for rolling bearing fault diagnosis under strong noises and variable loads. Ieee Access. 2020;8:66257-69.

33. Shin D-H, Chung K, Park RC. Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data. IEEE Access. 2020;8:150784-96.

34. Lee S, Xie K, Ngoduy D, Keyvan-Ekbatani M. An advanced deep learning approach to real-time estimation of lane-based queue lengths at a signalized junction. Transportation research part C: emerging technologies. 2019;109:117-36.

35. Li P, Abdel-Aty M, Yuan J. Real-time crash risk prediction on arterials based on LSTM-CNN. Accident Analysis & Prevention. 2020;135:105371.

36. Guo J, Xiong Q, Chen J, Miao E, Wu C, Zhu Q, et al. Study of static thermal deformation modeling based on a hybrid CNN-LSTM model with spatiotemporal correlation. The International Journal of Advanced Manufacturing Technology. 2022:1-13.

37. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press In; 2001.

38. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735-80.

39. Graves A, Mohamed A-r, Hinton G, editors. Speech recognition with deep recurrent neural networks. 2013 IEEE international conference on acoustics, speech and signal processing; 2013: Ieee.

40. Chen C, Kwon J, Rice J, Skabardonis A, Varaiya P. Detecting errors and imputing missing data for single-loop surveillance systems. Transportation Research Record: Journal of the Transportation Research Board. 2003(1855):160-7.

41. Aydilek IB, Arslan A. A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm. Information Sciences. 2013;233:25-35.

42. Tak S, Woo S, Yeo H. Data-Driven Imputation Method for Traffic Data in Sectional Units of Road Links. IEEE Trans Intelligent Transportation Systems. 2016;17(6):1762-71.