Accident Analysis and Prevention 144 (2020) 105658

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

⋆This paper has been handled by associate editor Tony Sze.The application of novel connected vehicles emulated data on real-time crash potential prediction for arterials

Pei Li*, Mohamed Abdel-Aty, Qing Cai, Cheng Yuan

Department of Civil, Environmental & Construction Engineering, University of Central Florida, Orlando, FL 32816, United States

ARTICLE INFO

Keywords:
Real-time crash potential prediction
Connected vehicle emulated data
Urban arterials
Deep learning

ABSTRACT

Real-time crash potential prediction could provide valuable information for Active Traffic Management Systems. Fixed infrastructure-based vehicle detection devices were widely used in the previous studies to obtain different types of data for crash potential prediction. However, it was difficult to obtain data in large range through these devices due to the costs of installation and maintenance. This paper introduced a novel connected vehicle (CV) emulated data for real-time crash potential prediction. Different from the fixed devices' data, CV emulated data have high flexibility and can be obtained continuously with relatively low cost. Crash and CV emulated data were collected from two urban arterials in Orlando, USA. Crash data were archived by the Signal for Analytics system (S4A), while the CV emulated data were obtained through the data collection API with a high frequency. Different data cleaning and preparation techniques were implemented, while various speed-related variables were generated from the CV emulated data. A Long Short-term Memory (LSTM) neural network was trained to predict the crash potential in the next 5 – 10 min. The results from the model illustrated the feasibility of using a novel CV emulated data to predict real-time crash potential. The average and 50th percentile speed were the two most important variables for the crash potential prediction. In addition, the proposed LSTM outperformed Bayesian logistics regression and XGBoost in terms of sensitivity, Area under Curve (AUC), and false alarm rate. With the rapid development of the connected vehicle systems, the results from this paper can be extended to other types of vehicles and data, which can significantly enhance traffic safety.

1. Introduction

In 2018, traffic crashes caused 33,654 fatalities in the USA (IIHS, 2019), while 41.6 % of them happened on urban arterials. Improving traffic safety, especially for urban arterials, is becoming a major concern for traffic engineers and researchers. Real-time crash potential prediction is one of the effective methods for enhancing traffic safety. Different from the traditional crash frequency prediction based on aggregated data, real-time crash potential prediction aims to predict the crash probability during a short-time interval. However, most of the existing studies on real-time crash potential prediction are limited to freeways (Abdel-Aty et al., 2012; Ahmed et al., 2012; Xu et al., 2013; Yu and Abdel-Aty, 2014a) rather than urban arterials (Wang et al., 2015b; Yuan and Abdel-Aty, 2018; Li et al., 2020). Urban arterials usually have more complicated traffic conditions, which require various data sources to predict the real-time crash potential, such as traffic, signal, and weather data. Traditional safety studies usually obtained these data from the fixed infrastructure-based devices, including loop detectors, Bluetooth detectors, microwave sensors, and cameras (Hassan and Abdel-Aty, 2013; Wang et al., 2015a). However, these devices require extra installation costs and regular maintenance. In addition, some devices, such as cameras, are sensitive to lighting, weather conditions, etc. Also, the detection range of these fixed devices are limited to their locations.

Recently, the concept of CV provides a novel way to obtain vehicle data in large scale with high flexibility and low cost. Different from the traditional sensor data, the CV data are easy to obtain and maintain. In addition, the data can be collected continuously in a wide range. It is possible to depict the traffic conditions of the whole city with large-scale vehicles. The real deployment of the CV system still needs more time. However, with the help of the mobile sensing technology, it is possible to obtain CV emulated data. CV emulated data can provide similar vehicle information as CV data, such as vehicle location, speed, etc. There are some studies that are related to the applications of the CV

* Corresponding author.
E-mail addresses: peili@knights.ucf.edu (P. Li), M.Aty@ucf.edu (M. Abdel-Aty), qingcai@Knights.ucf.edu (Q. Cai), yuancheng0124@knights.ucf.edu (C. Yuan).

https://doi.org/10.1016/j.aap.2020.105658
Received 15 March 2020; Received in revised form 26 May 2020; Accepted 17 June 2020
Available online 03 July 2020

emulated data in the transportation field, such as anomaly detection (Pang et al., 2013; Kuang et al., 2015), traffic conditions estimation (Herring et al., 2010; Rahmani et al., 2015), etc. Nevertheless, only few studies applied this new data source to the traffic safety field. Xie et al. (2013) used taxi data to calculate arterial-level travel speed and introduced speed as an explanatory variable to investigate intersection safety in Shanghai. The authors found higher average speeds along arterials were associated with increased intersection crashes. Similarly, Wang et al. (2015b) examined the relationship between different variables from taxi GPS data and traffic safety for urban arterials during peak and off-peak hours. Higher average speeds were found to be associated with higher crash frequencies during peak periods, but not during off-peak periods. Bao et al. (2019) used the numbers of taxi pick-ups and drop-offs as new variables to predict citywide crash frequency based on deep learning models. Wang et al. (2019) applied a support vector machine (SVM) model to predict crash potential on freeways based on taxi data. Different variables were generated, such as average speed, speed difference ratio, etc. In addition, SVM were found to have better performance than the logistic regression model in terms of sensitivity and Area Under Curve (AUC) values.

Two types of models are available for real-time crash potential prediction, statistical models and machine learning models. Statistical models include logistic regression, Bayesian logistics regression (Ahmed et al., 2012), etc. These models were usually built on matched-case control data and had certain assumptions. Considering these limitations, the applications of machine learning methods were explored, such as Support Vector Machine (SVM) (Yu and Abdel-Aty, 2013), Random Forest (Lin et al., 2015), etc. The performance of these methods was proven to be better than the statistical methods. For example, Yu and Abdel-Aty (2013) indicated SVM outperformed Bayesian logistic regression in terms of AUC value. Recently, the availability of massive transportation data and the development of computer hardware accelerate the implementation of deep learning. Deep learning is one class of machine learning methods. It was utilized to solve various transportation problems. Moreover, Recurrent Neural Network (RNN) was proven to be especially useful for learning time-series transportation data (Zhang et al., 2020). Different from the traditional neural network that only maps the current input vector to output vector, RNN introduces recurrent connections, which allow information to persist. However, one drawback of the RNN is that it cannot capture long-term dependencies (Hochreiter, 1991). Thus, long short-term memory neural network (LSTM), was invented by Hochreiter and Schmidhuber (1997). LSTM improves the performance of RNN by including memory cells and gates, which preserve the information for a long period. There are some new studies that applied LSTM in transportation safety. Yuan et al. (2019) utilized LSTM to predict crash potential in real-time, the authors claimed that their models achieved better sensitivity than the conditional logistic model. Bao et al. (2019) implemented a spatiotemporal convolutional LSTM to predict the citywide crash frequency based on multiple data sources, such as taxi trip data, road network attributes, and land use features.

There are still several research gaps that need to be filled. First, the existing traffic safety studies with CV emulated data mainly focused on crash frequency analysis (Xie et al., 2013; Wang et al., 2015b; Bao et al., 2019) rather than crash potential prediction (Wang et al., 2019). It is necessary to investigate the feasibility of using CV emulated data for real-time crash potential, especially for urban arterials. Second, almost all the studies utilized taxi for traffic safety analysis. More efforts need to be done on other types of vehicles. Basso et al. (2020) also indicate it is necessary to distinguish different types of vehicles for crash prediction. Previous studies successfully detected traffic anomaly based on bus data (Kong et al., 2017; Zhang et al., 2019). It is promising to investigate the applications of bus data on traffic safety.

Different from the other vehicles such as taxis, buses have their unique advantages. For example, a bus usually has fixed route and time. The trajectory of bus is more stable and cannot be affected by the drivers' preferences and characteristics. Moreover, bus usually runs around the urban area, which can depict the city-wide traffic conditions extensively. Third, the studied periods are restricted in the existing studies, which are not favorable for realistic applications. For example, Wang et al. (2015b) only analyzed the crashes during peak and off-peak periods. Wang et al. (2019) only selected the time periods from 5 to 10 minutes (and 10 − 15 min) prior to the events (crash and non-crash cases) to conduct analysis. Although the authors claimed the non-crash cases were randomly picked, the information of the unselected non-crash cases is still important to the model. It is necessary to build a generic model based on the entire data set.

The main objective of this paper is to explore the feasibility of utilizing novel CV emulated data to predict real-time crash potential for arterial road segments. Two major urban arterials in Orlando, FL are selected to conduct a case study. Various speed-related variables are generated from the CV emulated data. In addition, different data preparation, map-matching techniques will be explored. A deep learning methodology is proposed to predict the real-time crash potential with variables from the CV emulated data. The proposed method will be compared with different benchmark methods based on various evaluation metrics.

2. Data Preparation

2.1. Data description

Two data sets are used in this study, CV emulated data and crash data. The CV emulated data have three parts: vehicle trajectory data, routes data, and stops data. All of them are obtained by the data collection API from the DoubleMap. The API requests are made with an HTTP GET request, and the data are returned in JSON format. There are around 300 LYNX® buses and 50 UCF shuttles in the vehicle trajectory data. The data are collected in real-time and updated every three seconds. The geographical location, heading, stop, and ID of the vehicle can be obtained from this data (Table 1). In addition, the decimal points of 90 % latitude and longitude data range from 5 to 15, which can achieve up to the millimeter accuracy. The routes data provide the information of the active LYNX® routes, such as the route ID, route name, and the route stops. The stops data are complementary to the routes data and provide the geographical location, ID, and name of each stop.

The vehicle trajectory data have an extremely high update frequency, with almost 3,000,000 records are generating every day. Therefore, it is difficult to conduct safety analysis for the entire city. Two urban arterials (Fig. 1), are selected in this study considering road geometry, vehicle density, crash frequency, etc. The experimental time range is from April 2019 to July 2019. The target of this paper is the road segment, which is defined as the road facility between two consecutive intersections with a certain direction (Fig. 2). Each road segment has its unique ID. In total, the selected area has 126 arterial segments, 66 intersections, and 15 bus routes.

Table 1
Data description.

Name	Description
ID	Unique integer for each vehicle.
Name	A text name for the vehicle.
Latitude	Latitude for the current position of the vehicle.
Longitude	Longitude for the current position of the vehicle.
Heading	The direction of movement, in degrees (0 − 360).
Route	The ID of the route that this vehicle is currently assigned to.
Laststop	The ID of the stop that this vehicle was most recently at, or its current stop.
Lastupdate	The UNIX timestamp of the last GPS update from the vehicle.
Source	LYNX® or UCF.

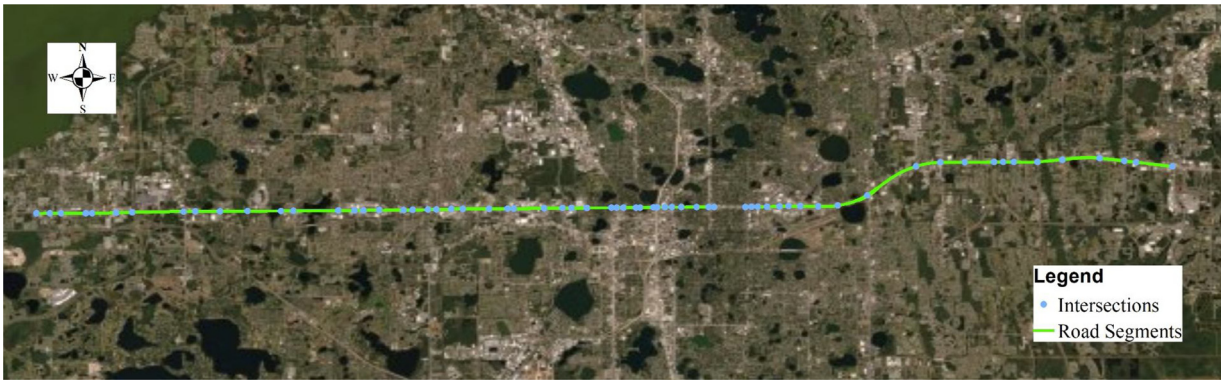


Fig. 1. Research area.

The crash data of the selected area are obtained from the S4A system. S4A provides detailed information for each crash event, including crash time, location, severity, and type.

2.2. Data preprocessing

The procedure of the CV emulated data preparation is shown in Fig. 3. One common drawback for using CV emulated data is the instability of the trajectory. Previous studies spent extensive time correcting the trajectory data based on different map-matching methods. The benefit from the bus is that it usually has a fixed route. In addition, we can obtain direction information to help the map matching process.

In terms of data preparation, first, the outliers and errors are removed from the original data, such as the GPS points outside of the studied area and the duplicated records. The obtained data are matched to the road segments for further analysis (Fig. 4). *ArcMap 10.6* is used for map matching. Since buses have fixed routes, the segment map is generated using a combination of bus routes and existing road network. This process greatly mitigates the computational costs compared with matching mixed trajectory to a large road network. Based on the map matching function provided by *ArcMap*, the CV emulated data are converted to the data of road segments. Moreover, the vehicle directions are also considered to improve the map matching accuracy.

Second, the speed of each vehicle is estimated based on two continuous GPS points. For a vehicle i, its speed at time t_1 is estimated as:

$$V_i = \frac{distance(GPS_{t_1},\ GPS_{t_2})}{t_1 - t_2} \tag{1}$$

Where GPS_{t_1} is its location at time t_1, and GPS_{t_2} is its location at time t_2. The distance d between two locations is estimated based on the Haversine formula (Veness, 2019):

$$a = sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\bullet\cos(\varphi_2)\bullet sin^2\left(\frac{\Delta\lambda}{2}\right) \tag{2}$$

$$c = 2\bullet arctan2(\sqrt{a},\ \sqrt{1-a}) \tag{3}$$

$$d = R\bullet c \tag{4}$$

Where φ and λ are latitude and longitude in radians, R is earth's radius. After we get the vehicle speed, several variables for the road segments can be generated, including average speed, speed standard deviation, 85th percentile speed, 50th percentile speed, and 15th percentile speed. All these variables are generated in 1-minute time interval. The descriptive table of the variables is shown in Table 2.

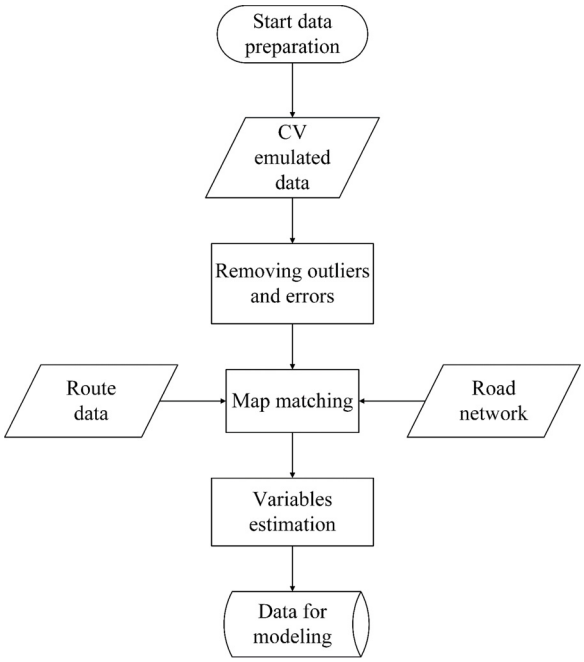


Fig. 3. Procedure of data preparation.

In addition, to illustrate the results from the speed estimation, the average bus speed of one road segment is shown in Fig. 5. The speed reaches two low points at 8:00 and 17:00, which correspond to the peak hours of daily commutes. The bus usually has a constant low speed from 8:00 to 17:00. The speed gradually increases after 17:00 and reaches the highest point around midnight.

To prepare the crash data, the crashes which are within the intersection influence area (within 250 feet of intersection) are excluded from the original data set. In addition, alcohol and drugs-related crashes are also removed. After these processes, there are 180 crashes in total. These crashes are then matched to the corresponding segments according to the geographical locations.

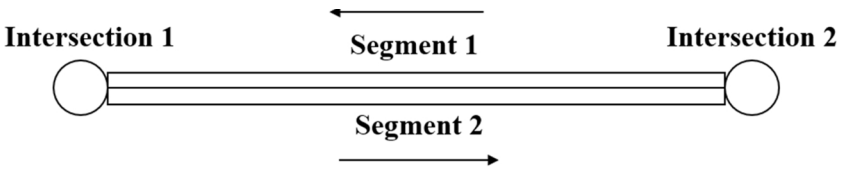
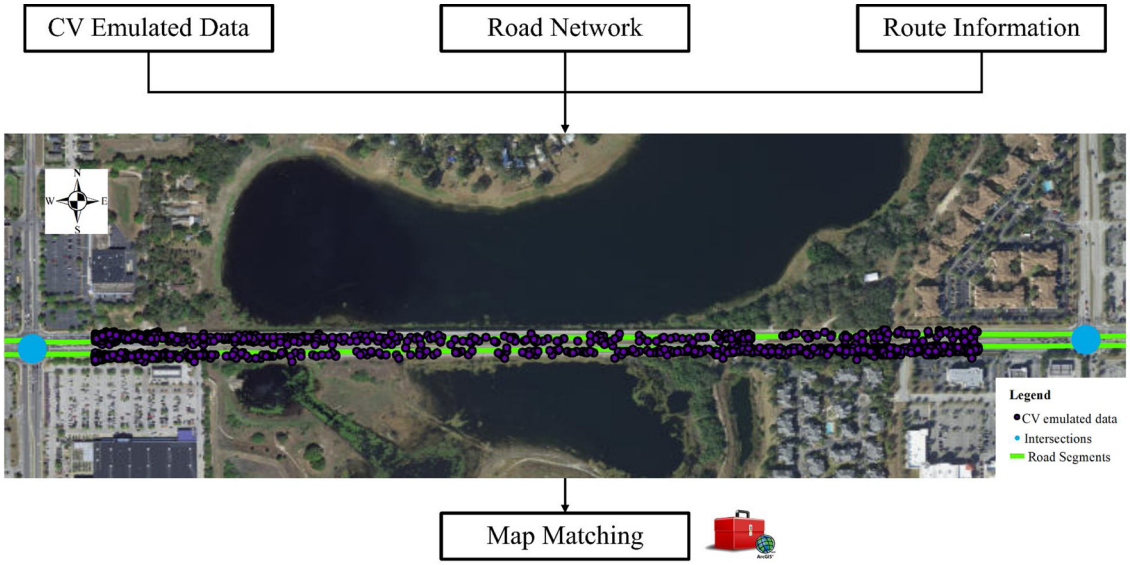


Fig. 2. Illustration of segments and intersections.



ID	Latitude	Longitude	…
17706218	28.548538	-81.381144	
17706218	28.549315	-81.379384	
17706218	28.549304	-81.379367	

ID	Latitude	Longitude	Road ID	…
17706218	28.548538	-81.381144	1779	
17706218	28.549315	-81.379384	1779	
17706218	28.549304	-81.379367	1779	

Fig. 4. Map matching process.

Table 2
Descriptive Statistics.

Variable Name	Description	Mean	Std	Min	Max
Avg_speed	Average speed of the segment	15.91	11.70	0.00	50.85
Std_speed	Speed standard deviation of the segment	9.37	4.71	0.00	28.42
Per85	85th percentile speed of the segment	21.88	8.80	0.00	49.03
Per50	50th percentile speed of the segment	15.33	9.49	0.00	47.64
Per15	15th percentile speed of the segment	8.58	8.40	0.00	47.64

Note: all the values are in the unit of mph.

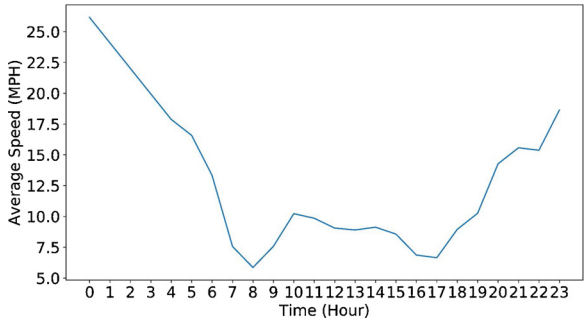


Fig. 5. Average segment speed.

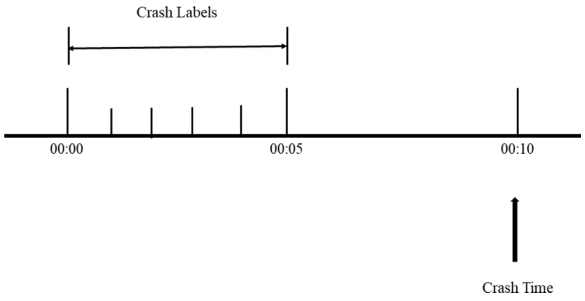


Fig. 6. Crash labelling process.

2.3. Crash labelling and features selection

This paper aims to utilize the features from the CV emulated data to predict real-time crash potential. After the vehicle features and crashes are matched to the corresponding road segments, data of each segment are prepared in chronological order. The time range of the data is from 4/16/2020 to 7/2/2020. Therefore, each road segment has roughly 112,320 records. This paper aims to predict the crash potential in the next 5 − 10 min. As Fig. 6 shows, if a crash happened at 00:10, the traffic safety statuses from 00:00 to 00:05 are labelled as '1', indicating that a crash will occur in the next 5 − 10 min from the current time. Otherwise, the traffic safety status is labeled as '0'. In addition, since a crash event usually causes turbulence to the traffic conditions, data within 120 min after a crash are removed.

The variables used by this paper were commonly applied by the previous traffic safety studies. The average speed and speed standard deviation were proven to be closely related to crash potential (Yu and Abdel-Aty, 2013, 2014b). Park and Saccomanno (2006) indicated it is necessary to include 85th percentile speed for crash risk evaluation. Muchuruza and Mussa (2005) pointed out the missing of low speed vehicles among the existing studies. The authors introduced 15th percentile speed as a new variable for traffic safety analysis. In addition, acceleration may also be utilized as an indicator for crash. Stipancic

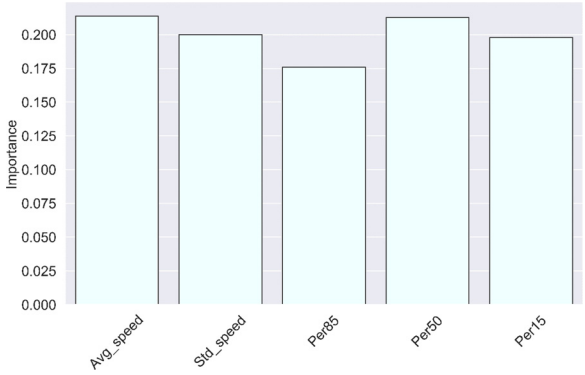


Fig. 7. Variables' importance.

et al. (2018) introduced hard braking and accelerating events as surrogate safety measures to investigate their correlations with historical collisions. However, acceleration is usually calculated in high frequency and this paper is using the 1-min interval. In the end, average speed, speed standard deviation, 85th percentile speed, 50th percentile speed, and 15th percentile speed are selected in this paper. To illustrate the feasibility of using these variables, the variables' importance is estimated based on the Extra-tree classifier (Geurts et al., 2006). Extra-tree classifier fits a number of randomized decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting (Pedregosa et al., 2011). The variables importance of the Table 2 is shown in Fig. 7. The average speed is the most important variable for crash potential prediction. While the 50th percentile speed is the second important variable. The speed standard deviation has similar importance as the 15th percentile speed. Based on the results from Fig. 7 and existing studies, this paper utilizes all these five variables to predict crash potential.

3. Methodologies

In this section, we mainly present the LSTM model used in this paper. LSTM is one type of RNNs. The basic idea of RNN is having loops inside to process the past information for future prediction. The RNN has chain-like design to allow the past information to be processed, which is shown in Fig. 8. The unique architecture of RNN enables its good performance on sequence and time-series data.

However, during the process of backpropagation, the gradients of RNN usually become vanishingly small over long distances. This drawback makes RNN unable to handle the long-term dependencies (Hochreiter, 1991). Therefore, Hochreiter and Schmidhuber (1997) proposed LSTM to solve the vanishing gradients problem, which introduced the memory cells to determine when to forget certain information.

The structure of an LSTM layer is shown in Fig. 9. LSTM has three different gates compared with the common RNN (Fig. 8). Specifically, the forget gate f_t controls how much to forget from the previous step memory cell. The input gate i_t determines which values to be updated. The output gate o_t determines the output for the hidden state h_t. The introduction of these special gates makes it easier for LSTM to preserve information over long timestamps. For instance, if we take the forget gate as 1 and the input gate as 0, then the information of this memory cell is preserved indefinitely.

If the input of LSTM is denotes as $X = (X_1, X_2, ...,X_t)$, where t is the prediction period, $h = (h_1, h_2, ...,h_t)$ is the hidden state, and the $y = (y_1, y_2, ...,y_t)$ is the output. The essential equations for LSTM are shown from Eq. (5)–(10) (Kang et al., 2017).

$$\boldsymbol{i_t} = \boldsymbol{\sigma}(\boldsymbol{W_{ix}X_t} + \boldsymbol{W_{ih}h_{t-1}} + \boldsymbol{W_{ic}c_{t-1}} + \boldsymbol{b_i}) \tag{5}$$

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_{fx}X_t} + \boldsymbol{W_{hf}h_{t-1}} + \boldsymbol{W_{cf}c_{t-1}} + \boldsymbol{b_f}) \tag{6}$$

$$\boldsymbol{o_t} = \boldsymbol{\sigma}(\boldsymbol{W_{ox}X_t} + \boldsymbol{W_{oh}h_{t-1}} + \boldsymbol{W_{oc}c_t} + \boldsymbol{b_o}) \tag{7}$$

$$\boldsymbol{c_t} = \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \odot \boldsymbol{tanh}(\boldsymbol{W_{cx}X_t} + \boldsymbol{W_{ch}h_{t-1}} + \boldsymbol{b_c}) \tag{8}$$

$$\boldsymbol{h_t} = \boldsymbol{o_t} \odot \boldsymbol{tanh}(\boldsymbol{c_t}) \tag{9}$$

$$\boldsymbol{y_t} = \boldsymbol{W_{yh}h_{t-1}} + \boldsymbol{b_y} \tag{10}$$

Where W represents weight matrices, for example, W_{ix} denotes the weight matrix from the input gate to the input, σ is the logistic sigmoid function and $\odot$ indicates elementwise product of the vectors.

4. Experimental design and results

4.1. Experimental design

The procedure of the experiment is shown in Fig. 10. The data are first divided into training (75 %) and test (25 %). As crashes are rare events, the data are highly imbalanced. The ratio of non-crash events to crash events is around 9,000:1 in the training data. Directly applying the model to the training data will result in a model with bad performance. Therefore, data resampling method should be implemented before training the model. Matched-case control is a traditional under-sampling method, which creates non-crash events based on several control factors from the crash events, such as the time of the day, day of the week, etc. However, the information of the non-crash events may be lost during this process. Synthetic Minority Over-sampling Technique (SMOTE) is a popular data resampling method (Chawla et al., 2002) and has been widely applied to similar problems. SMOTE is a data over-sampling method which synthesizes new minority samples between existing minority samples. The advantage of the SMOTE is the number of majority samples will not change, which retains their information completely. After the training data is resampled by SMOTE, the

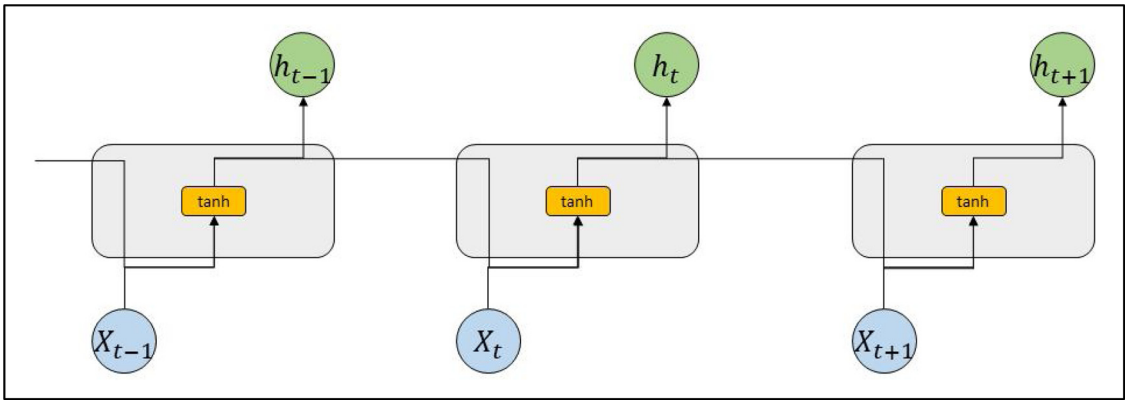


Fig. 8. RNN structure.

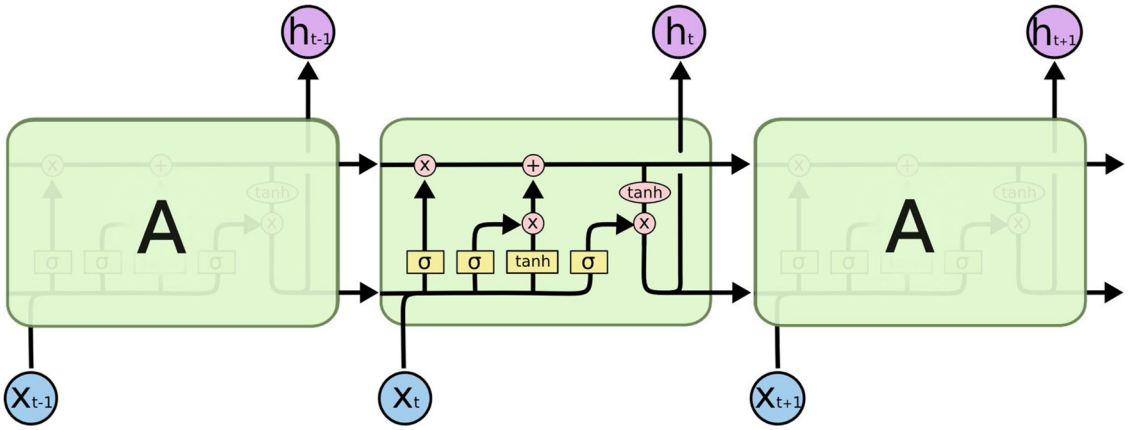


Fig. 9. LSTM structure (Olah, 2015).

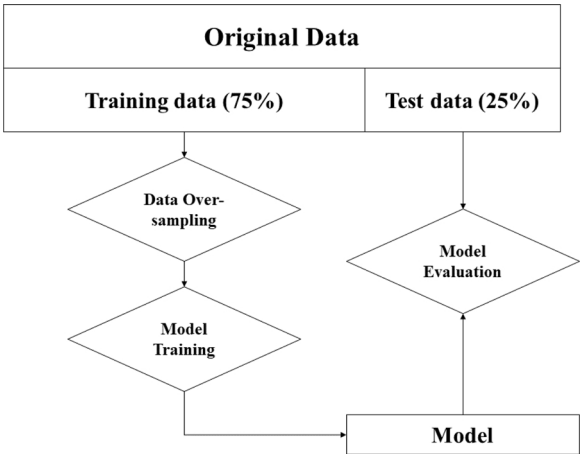


Fig. 10. Experiment procedures.

proposed model is trained on the balanced data. Finally, the model is evaluated on the test data with different metrics.

In terms of the evaluation metrics, sensitivity, false alarm rate, and AUC value are applied. Sensitivity and false alarm rate are estimated according to the confusion matrix (Table 3) and Eq.s (11)–(12). High sensitivity means the model can predict most of the crash cases correctly, while a low false alarm rate indicates the model predicts most non-crash cases correctly. AUC is a comprehensive metric which is estimated based on the Receiver Operating Characteristic (ROC) curve. The ROC curve plots true positive rate and false positive rate, which can be estimated from Eq.s (11) and (12). AUC measures the entire two-dimensional area underneath the entire ROC curve from (0, 0) to (1, 1).

$$\textbf{True Postive Rate (Sensitivity)} = \frac{TP}{TP + FN} \tag{11}$$

$$\textbf{False Postive Rate (False Alarm Rate)} = \frac{FP}{FP + TN} \tag{12}$$

The LSTM used in this paper are designed to take a three-dimensional data as the input and generate a single output. Due to the special requirement of the LSTM, the input data are reshaped with a dimension as (sample size * time step * feature number). Specifically, the data from the previous T-10 min are stacked together as the input to predict the crash potential at the time T. The dropout layer is introduced to prevent over-fitting. Sigmoid function is used to generate the output since crash prediction is a binary classification problem. The network architecture of the neural network is shown in Fig. 11.

Table 3
Confusion matrix.

	True Crash	True Non-Crash
Predicted Crash	True Positive (TP)	False Positive (FP)
Predicted Non-Crash	False Negative (FN)	True Negative (TN)

4.2. Experimental results

The performance of the deep learning models is heavily dependent on hyperparameters. Six common hyperparameters are selected in this paper, including LSTM unit number, dropout rate, optimization function, learning rate, epoch number, and batch size. The proposed model is implemented on the Keras (Chollet, 2015) framework. The hyperparameters are tuned by iterating all the permutations of the given values from Table 4. After each iteration, the model is evaluated on the test dataset with AUC value. The model with the highest AUC is selected, and the best combination of hyperparameters is also shown in Table 4.

The process of hyperparameters usually takes a lot of time since there are many permutations. However, several common conclusions can be summarized based on the tuning results. First, a higher LSTM unit number may result in an over-fitting model and impair the performance on the test data. Second, the learning rate determines the process of optimization. Setting the high learning rate too high can let the model miss the global minimum and end up with the local minimum. However, a too low learning rate may increase the training time. Third, batch size is the number of training examples in one iteration. Mini-batch is a relatively desirable choice since it could achieve a relative speedy convergence and is more computationally efficient.

The experiment results of the model on the test data is shown in Table 5. The model could successfully predict 79 % of the total crashes, with a relatively low false alarm rate as 21 %. In addition, the confusion matrix of the model is shown in Fig. 12. These results illustrate the feasibility of using CV emulated data to predict crash potential in real-time.

In addition, the t-SNEs of the raw data and extracted features are shown in Fig. 13 (a) and Fig. 13 (b), respectively. t-SNEs is an effective data visualization technique, especially for high-dimensional data (Maaten and Hinton, 2008). The extracted features are obtained through the last layer of the LSTM. Crash and non-crash events are extremely tangled together and hard to distinguish in the raw data. On the contrary, the features extracted by the LSTM (Fig. 13 (b)) successfully separate these two events, which illustrate the good performance of the model.

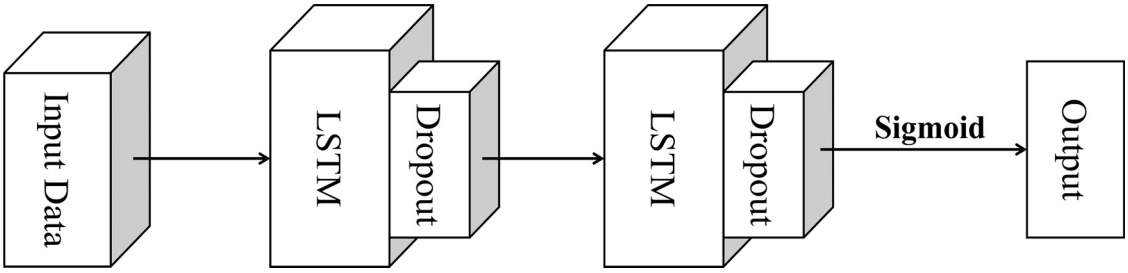


Fig. 11. Network architecture.

Table 4
Hyperparameters tuning.

Name	Range	Value
LSTM unit number	128, 64, 32, 16	32
Dropout Rate	0.3, 0.4, 0.5	0.5
Optimization Function	SGD, Adam, RMSprop	RMSprop
Learning rate	0.1, 0.01, 0.001, 0.0001	0.001
Batch size	1000, 5000, 10,000	5000
Epoch number	50, 100, 150, 200	50

Table 5
Model results.

Name	Value
AUC	0.79
Sensitivity	0.79
False Alarm Rate	0.21

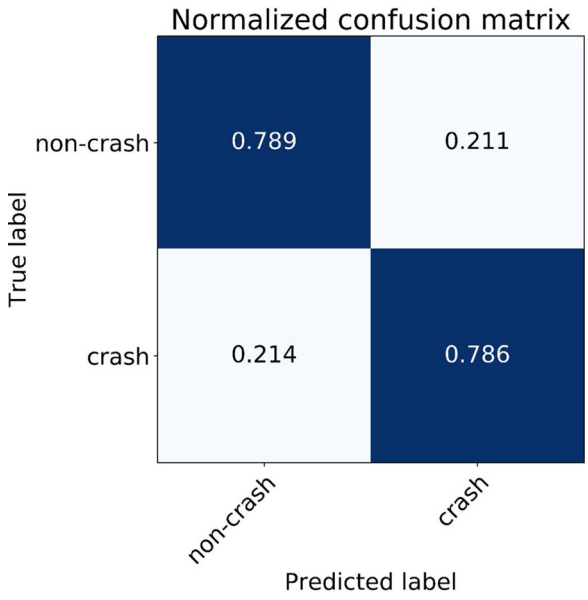


Fig. 12. Confusion matrix.

4.3. Model comparison

This section compares the performance of the proposed LSTM with two types of benchmark methods, statistical methods and machine learning methods. Bayesian logistics regression is used as the statistical method. This method was widely applied on the real-time crash potential prediction (Ahmed et al., 2012; Yu and Abdel-Aty, 2013). Different from the basic logistics regression, in the Bayesian approach, the parameters are treated as random variables and the data are used to update beliefs about the behavior of the parameters to assess their distributional properties (Ahmed et al., 2012). For machine learning methods, eXtreme Gradient Boosting (XGBoost) is selected due to its good performance on the similar problems (Chen and Guestrin, 2016). These two methods are trained based on the same dataset as the LSTM. Several hyperparameters of XGBoost are tuned, such as the number of trees, maximum tree depth, etc. The results of model comparison are shown in Fig. 14. LSTM outperforms other two methods in terms of sensitivity, false alarm rate, and AUC.

5. Conclusions

This paper applied a new CV emulated data source to predict real-time crash potential for urban arterials. Two urban arterials in Orlando, USA were selected to conduct a case study. Crash and CV emulated data were obtained for three months. The CV emulated data were used to generated different speed-related variables, such as average speed, speed standard deviation, 85th percentile speed, etc. After data cleaning and preparation, an LSTM model was proposed to predict the crash potential in the next 5 − 10 min. The model was trained on the data after over-sampling and validated based on different evaluation metrics. Results illustrated the feasibility of using CV emulated data for real-time crash potential. In addition, the model comparison results indicated the LSTM outperformed other methods, including Bayesian logistics regression and XGBoost in terms of sensitivity, false alarm rate, and AUC. Several key findings from this paper can be summarized as below:

- Different speed-related variables can be generated from the CV emulated data, such as the average speed, speed standard deviation, 85th percentile speed, etc. These variables can be updated in a high frequency to better reflect the continuous traffic conditions.
- With the help of map-matching methods, the variables from bus data can be transformed to the variables of the road segments. Different from other types of vehicles such as taxis, buses usually have fixed routes and schedule. The computation costs of map matching are greatly mitigated for bus data.
- The CV emulated data can be used as a new data source to predict real-time crash potential. The data have higher flexibility and wider coverage compared with the traditional sensor data. With the rapid development of the CV, the results from this paper can be generalized to other types of vehicles.
- LSTM outperforms other types of models in terms of sensitivity, false alarm rate, and AUC values. With the help of the over-sampling methods, the LSTM can learn time-series data comprehensively. In addition, due to the rapid development of the computer hardware, the implementations of the deep learning models will be much easier in the future.

There are several possible applications based on the results of this paper. For example, the proposed model can be deployed in a cloud server, taking the real-time vehicles' location data and generating the crash potential. The results can then be visualized on an interactive map and provide safety notifications for traffic managers and decision

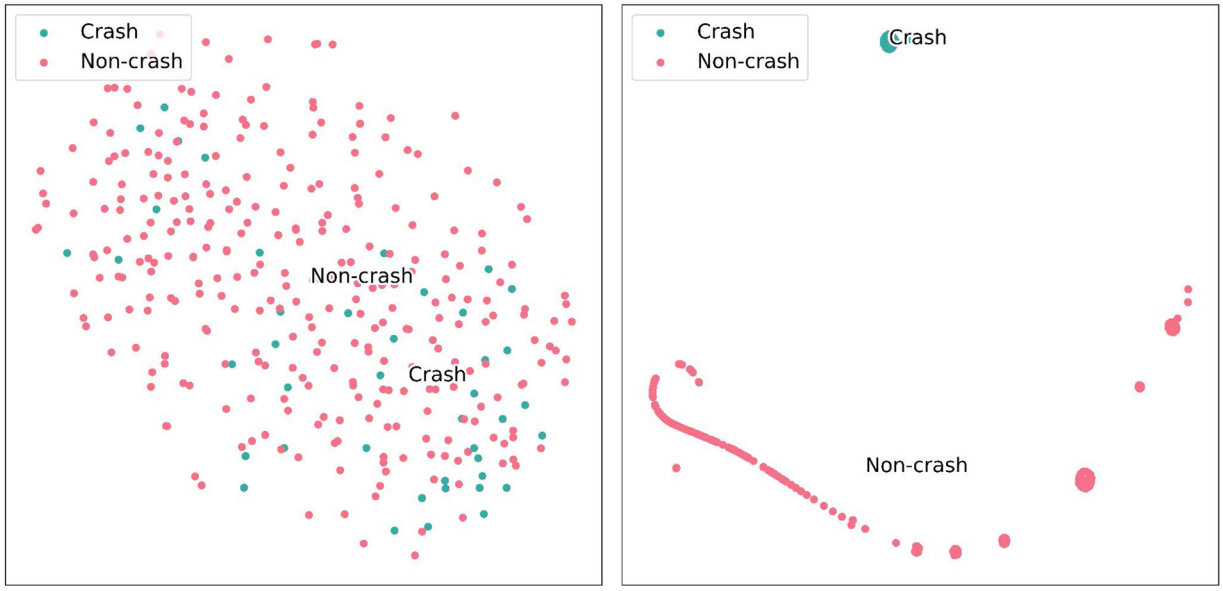


Fig. 13. t-SNEs of the raw data and extracted features.

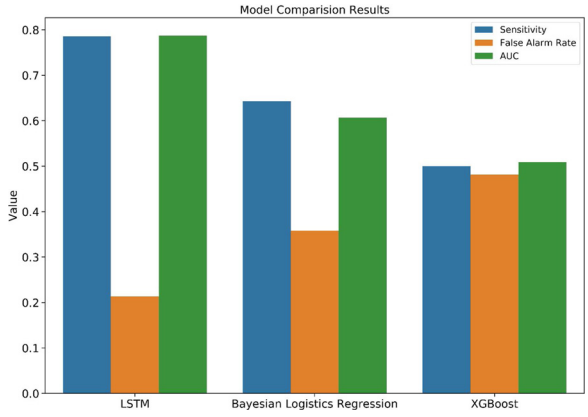


Fig. 14. Model comparison results.

makers. Moreover, safety information can also be transmitted to the drivers who are connected. The drivers can take corresponding actions. In addition, since the data used in this paper have similar structure as CV data. The methods can be easily generalized to CV data when available, which could help enhance traffic safety on a large-scale.

There are still several improvements that can be done in the future. First, buses are one type of vehicles. It is very promising to explore the fusion with other types vehicles, such as taxis, private vehicles, trucks, etc. Second, the impact of the different variables on crash potential prediction also needs further investigation, a proper variables generation and selection process could possibly improve the performance of the model. Forth, different deep learning architectures can be explored in the future to improve the results of the current model. Finally, it would be promising to combine the results from this paper with other similar studies. For example, Wiseman and Grinberg (2016) proposed a real-time crash potential damages assessment approach for autonomous vehicles. If an autonomous vehicle can receive the crash potential prediction results through CV as suggested in our paper, the information may help it to avoid certain crashes. For the case of inevitable crash, the crash potential damages assessment can help the vehicle achieve the least damages.

CRediT authorship contribution statement

Pei Li: Conceptualization, Methodology, Software, Visualization, Formal analysis, Validation, Writing - original draft, Writing - review & editing. **Mohamed Abdel-Aty:** Conceptualization, Methodology, Validation, Supervision, Writing - review & editing. **Qing Cai:** Conceptualization, Methodology. **Cheng Yuan:** Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdel-Aty, M.A., Hassan, H.M., Ahmed, M., Al-Ghamdi, A.S., 2012. Real-time prediction of visibility related crashes. Transp. Res. Part C Emerg. Technol. 24, 288–298.
Ahmed, M.M., Abdel-Aty, M., Yu, R., 2012. Assessment of interaction of crash occurrence, mountainous freeway geometry, real-time weather, and traffic data. Transp. Res. Rec. 2280 (1), 51–59.
Bao, J., Liu, P., Ukkusuri, S.V., 2019. A spatiotemporal deep learning approach for city-wide short-term crash risk prediction with multi-source data. Accid. Anal. Prev. 122, 239–254.
Basso, F., Basso, L.J., Pezoa, R., 2020. The importance of flow composition in real-time crash prediction. Accid. Anal. Prev. 137, 105436.
Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.
Chen, T., Guestrin, C., 2016. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery. San Francisco, California, USA. pp. 785–794.
Chollet, F., 2015. Keras.
Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63 (1), 3–42.
Hassan, H.M., Abdel-Aty, M.A., 2013. Predicting reduced visibility related crashes on freeways using real-time traffic flow data. J. Safety Res. 45, 29–36.
Herring, R., Hofleitner, A., Abbeel, P., Bayen, A., 2010. Estimating arterial traffic conditions using sparse probe data. Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems 929–936.
Hochreiter, S., 1991. Untersuchungen zu dynamischen neuronalen netzen. Technische Universität München.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Iihs, 2019. Fatality Facts 2018 urban/rural Comparison.
Kang, D., Lv, Y., Chen, Y., 2017. Short-term traffic flow prediction with lstm recurrent neural network. Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) 1–6.

Kong, X., Song, X., Xia, F., Guo, H., Wang, J., Tolba, A., 2017. Lotad: long-term traffic anomaly detection based on crowdsourced bus trajectory data. World Wide Web 21 (3), 825–847.
Kuang, W., An, S., Jiang, H., 2015. Detecting traffic anomalies in urban areas using taxi gps data. Math. Probl. Eng. 2015, 1–13.
Li, P., Abdel-Aty, M., Yuan, J., 2020. Real-time crash risk prediction on arterials based on lstm-cnn. Accid. Anal. Prev. 135, 105371.
Lin, L., Wang, Q., Sadek, A.W., 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. Transp. Res. Part C Emerg. Technol. 55, 444–459.
Maaten, L.V.D., Hinton, G., 2008. Visualizing data using t-sne. J. Mach. Learn. Res. 9 (November), 2579–2605.
Muchuruza, V., Mussa, R., 2005. Traffic operation and safety analyses of minimum speed limits on florida rural interstate highways. Proceedings of the Proceedings of the 2005 Mid-Continent Transportation Research Symposium 1–10.
Olah, C., 2015. Understanding Lstm Networks.
Pang, L.X., Chawla, S., Liu, W., Zheng, Y., 2013. On detection of emerging anomalous traffic patterns using gps data. Data Knowl. Eng. 87, 357–373.
Park, Y.-J., Saccomanno, F.F., 2006. Evaluating speed consistency between successive elements of a two-lane rural highway. Transp. Res. Part A Policy Pract. 40 (5), 375–385.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12 (October), 2825–2830.
Rahmani, M., Jenelius, E., Koutsopoulos, H.N., 2015. Non-parametric estimation of route travel time distributions from low-frequency floating car data. Transp. Res. Part C Emerg. Technol. 58, 343–362.
Stipancic, J., Miranda-Moreno, L., Saunier, N., 2018. Vehicle manoeuvers as surrogate safety measures: extracting data from the gps-enabled smartphones of regular drivers. Accid. Anal. Prev. 115, 160–169.
Veness, C., 2019. Calculate Distance, Bearing and More Between latitude/longitude Points.
Wang, L., Abdel-Aty, M., Shi, Q., Park, J., 2015a. Real-time crash prediction for expressway weaving segments. Transp. Res. Part C Emerg. Technol. 61, 1–10.
Wang, X., Fan, T., Chen, M., Deng, B., Wu, B., Tremont, P., 2015b. Safety modeling of urban arterials in shanghai, china. Accid. Anal. Prev. 83, 57–66.
Wang, J., Luo, T., Fu, T., 2019. Crash prediction based on traffic platoon characteristics using floating car trajectory data and the machine learning approach. Accid. Anal. Prev. 133, 105320.
Wiseman, Y., Grinberg, I., 2016. Circumspectly crash of autonomous vehicles. Proceedings of the 2016 IEEE International Conference on Electro Information Technology (EIT) 0387–0392.
Xie, K., Wang, X., Huang, H., Chen, X., 2013. Corridor-level signalized intersection safety analysis in shanghai, china using bayesian hierarchical models. Accid. Anal. Prev. 50, 25–33.
Xu, C., Tarko, A.P., Wang, W., Liu, P., 2013. Predicting crash likelihood and severity on freeways with real-time loop detector data. Accid. Anal. Prev. 57, 30–39.
Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. Accid. Anal. Prev. 51, 252–259.
Yu, R., Abdel-Aty, M., 2014a. Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data. Saf. Sci. 63, 50–56.
Yu, R., Abdel-Aty, M., 2014b. Using hierarchical bayesian binary probit models to analyze crash injury severity on high speed facilities with real-time traffic data. Accid. Anal. Prev. 62, 161–167.
Yuan, J., Abdel-Aty, M., 2018. Approach-level real-time crash risk analysis for signalized intersections. Accid. Anal. Prev. 119, 274–289.
Yuan, J., Abdel-Aty, M., Gong, Y., Cai, Q., 2019. Real-time crash risk prediction using long short-term memory recurrent neural network. Transp. Res. Rec. 2673 (4), 314–326.
Zhang, X., Zhang, X., Verma, S., Liu, Y., Blumenstein, M., Li, J., 2019. Detection of anomalous traffic patterns and insight analysis from bus trajectory data. Pricai 2019: Trends in Artificial Intelligence. pp. 307–321.
Zhang, S., Abdel-Aty, M., Yuan, J., Li, P., 2020. Prediction of pedestrian crossing intentions at intersections based on long short-term memory recurrent neural network. Transp. Res. Rec. 2674 (4), 57–65.