Accident Analysis and Prevention 141 (2020) 105520

Contents lists available at ScienceDirect

Accident Analysis and Prevention

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

A long short-term memory-based framework for crash detection on freeways with traffic data of different temporal resolutions

Feifeng Jiang*, Kwok Kit Richard Yuen, Eric Wai Ming Lee

Department of Architecture and Civil Engineering, City University of Hong Kong, Hong Kong, China

ARTICLE INFO

Keywords:
Crash detection
Deep learning methods
Long short-term memory networks
Traffic condition
Different temporal resolutions

ABSTRACT

Traffic crash detection is a major component of intelligent transportation systems. It can explore inner relationships between traffic conditions and crash risk, prevent potential crashes, and improve road safety. However, there exist some limitations in current studies on crash detection: (1) The commonly used machine learning methods cannot simulate the evolving transitions of traffic conditions before crash occurrences; (2) Current models collected traffic data of only one temporal resolution, which cannot fully represent traffic trends in different time intervals. Therefore, this study proposes a Long short-term memory (LSTM) based framework considering traffic data of different temporal resolutions (LSTMDTR) for crash detection. LSTM is an effective deep learning method to capture the long-term dependency and dynamic transitions of pre-crash conditions. Three LSTM networks considering traffic data of different temporal resolutions are constructed, which can comprehensively indicate traffic variations in different time intervals. A fully-connected layer is used to combine the outputs of three LSTM networks, and a dropout layer is used to reduce overfitting and improve prediction performance. The LSTMDTR model is implemented on datasets of I880-N and I805-N in California, America. The results indicate that the LSTMDTR model can obtain satisfactory performance on crash detection, with the highest crash accuracy of 70.43 %. LSTMDTR models constructed on one freeway can be transferred to other similar freeways, with 65.12 % of crash accuracy on transferability. Compared with machine learning methods and LSTM models with one or two temporal resolutions, the LSTMDTR model has been validated to perform better on crash detection and transferability. A proper number of neurons in the LSTMDTR model should be determined in real applications considering acceptable detection performance and computation time. The dropout technique can reduce overfitting and improve the generalization ability of the LSTMDTR model, increasing crash accuracy from 64.49 % to 70.43 %.

1. Introduction

Urban freeways are fundamental components of road transportation systems, facilitating long-distant and fast transportation among cities. However, due to fast speeds and turbulent traffic conditions, traffic crashes occur with high frequency on freeways (Pande and Abdel-Aty, 2006; Yuan and Abdel-Aty, 2018). They ultimately cause traffic congestion, property losses, and casualties. As reported by the National Highway Traffic Safety Administration in the U.S. (NHTSA), more than 630 million vehicle crashes occurred on freeways in 2015, 28 % of which led to physical injuries and fatalities (National Highway Traffic Safety Administration, 2016). Traffic crashes caused 242 billion economic losses and 25 % of traffic congestion on freeways (Federal Highway Administration, 2004; National Highway Traffic Safety Administration, 2016). However, traffic crashes are predictable and preventable by effective actions. Therefore, it is important and critical to detect and prevent traffic crashes on freeways to reduce casualties and improve road safety.

With the fast development of intelligent transportation systems (ITS), it becomes more efficient and convenient to collect and store traffic data from advanced sensing networks (e.g., loop detectors, radar techniques) (Hossain and Muromachi, 2012; Yu et al., 2016). Some studies have been conducted to detect traffic crashes by traffic data from ITS (Parsa et al., 2019; Yu and Abdel-Aty, 2013). They concluded that there existed a close relationship between traffic conditions and crash risk (Hossain and Muromachi, 2012; Sun and Sun, 2015; Xu et al., 2013). Therefore, "when and where a crash will occur" can be predicted by traffic conditions in advance. This can improve traffic safety on freeways by implementing proactive prevention strategies in real applications. For example, drivers can be reminded of potential crashes by

* Corresponding author.
E-mail address: feifjiang2-c@my.cityu.edu.hk (F. Jiang).

https://doi.org/10.1016/j.aap.2020.105520
Received 5 December 2019; Received in revised form 1 March 2020; Accepted 21 March 2020
Available online 08 April 2020

warning systems such as Dynamic Message Signs or in-vehicle devices (Sun et al., 2014). Traffic control and rescue devices can be arranged in advance to reduce the risk of traffic congestion and physical injuries (Sun and Sun, 2015).

After years of studies, some prediction models have been developed for crash detection with traffic data on freeways. The modeling methods can roughly be divided into two categories: statistical methods and machine learning methods (Hossain and Muromachi, 2012). Statistical methods contain logistic regression, log-linear model, etc. (Abdel-Aty et al., 2004; Lee et al., 2003). These statistical methods are commonly based on a linear assumption. However, the relationship between traffic conditions and crash risk is complicated and non-linear (Hossain and Muromachi, 2012; Sun and Sun, 2015). Therefore, machine learning methods designed for non-linear problems are commonly applied in current studies for crash detection, including logistic regression (LR), support vector machine (SVM), k-Nearest Neighbor (KNN), random forest (RF), etc (Abdel-Aty et al., 2012; Ahmed and Abdel-Aty, 2012; Liu et al., 2014; Xiao, 2019; Yao et al., 2014; Yuan and Cheu, 2003). For example, Xiao (2019) developed an SVM and KNN ensemble model for crash detection. Hossain and Muromachi (2012) constructed a Bayesian network-based framework to detect crashes on expressways.

However, the occurrences of crashes are induced by disturbance of traffic conditions (Hossain and Muromachi, 2012; Sun and Sun, 2015; Sun et al., 2014). It is critical for detection models to capture the dynamic transition process of pre-crash conditions with time-series traffic data. One limitation in current studies is that the classical machine learning methods cannot simulate the evolving transitions of traffic states before crash occurrences. This may limit their performance on crash detection. Some deep learning methods are state-of-the-art techniques for time-series problems, such as recurrent neural networks (RNN) and Long short-term memory (LSTM). However, few studies applied deep learning methods on crash detection, which have been validated to have better performance in other traffic domains (Lv et al., 2015). Therefore, a deep learning method that can simulate dynamic transitions of traffic conditions should be developed to improve the performance of crash prediction.

Another limitation of the current studies is that crash detection models are constructed with traffic data of only one temporal resolution. For example, Oh et al. (2005) developed a crash detection model with traffic data of 30 min. before crash occurrence. Sun and Sun (2015) used traffic data of 15 min. before crash occurrence. Sun et al. (2014) collected traffic data of 5–10 min. before a crash. However, traffic data of large temporal resolution (e.g., 30 min.) only indicates the traffic trends in a long-time-interval. And traffic data of small temporal resolution (e.g., 5 min.) only displays traffic variation within a short-time-interval. Traffic data of only one temporal resolution cannot fully represent traffic trend and dynamic transitions in different time intervals. This may affect the performance of crash detection models. Therefore, it is essential to develop a model considering traffic data of different time resolutions to improve prediction performance.

To address the above research gaps and limitations, this study proposes an LSTM-based framework considering traffic data of different temporal resolutions (LSTMDTR) for crash detection on freeways. LSTM is a deep learning method for time-series problems. It can learn long-term dependencies in sequential data and capture the dynamic transition process of pre-crash conditions. Three LSTM networks considering traffic data of different temporal resolutions are constructed. This can comprehensively indicate traffic trends and variations in different time intervals. A fully-connected layer is used to combine the outputs of three LSTM networks, and a dropout layer is used to reduce overfitting and improve prediction performance in complicated networks. The proposed LSTMDTR framework is conducted on a case study of I880-N in California, America for crash detection with different evaluation criteria. The model transferability is also examined on a case study of I805-N to evaluate if the detection model established on one freeway can be transferred to another similar freeway. The proposed framework is compared with other machine learning methods and LSTM models with one or two temporal resolutions to validate its effectiveness on crash detection and transferability. The effect of important parameters in the LSTMDTR model is analyzed and discussed in this study.

The main contribution of this study can be summarized as follows:

(1) Few studies applied deep learning methods on crash detection, which have been validated to have better performance in other traffic domains. This study is one of the few papers to explore the application of deep learning methods on crash detection.
(2) This paper is the first one to consider traffic data of different temporal resolutions. This can comprehensively indicate traffic trends and variations in different time intervals and improve prediction performance.

The remainder of this study is organized as follows: Section 2 details the proposed framework; Section 3 introduces data collection and preprocessing of the case study; Results and discussions are displayed in Section 4 and Section 5; Section 6 summarizes the conclusions to present the contributions and future work of this study.

2. Methodology

This study proposes a framework based on LSTM networks for crash detection with traffic data of different temporal resolutions. As Fig. 1 shows, the framework consists of three main parts: (1) data collection and preprocessing; (2) model development: LSTM-based framework considering traffic data of different temporal resolutions (LSTMDTR); (3) model evaluation. In the first part, both crash samples and non-crash samples with corresponding traffic data are collected and preprocessed to construct a dataset for crash detection. In the second part, the proposed LSTMDTR model is applied to the dataset to classify crash samples and non-crash samples. In the third part, four criteria are applied in this study to evaluate the performance of the LSTMDTR model on crash detection.

2.1. Data collection and preprocessing

The main purpose of this study is to explore the inner relationship between traffic conditions and crash risk on freeways. Traffic crashes can be predicted in advance by traffic conditions. Therefore, both crash cases and non-crash cases with corresponding traffic data should be collected. Traffic data contains information on speed, flow, and occupancy.

It should be noted that this study adopts a matched case-control method to collect non-crash cases, which is a commonly used method in current studies (Abdel-Aty et al., 2004; Hossain and Muromachi, 2012). For each crash case, a certain number of non-crash cases are randomly selected in consideration of the same time, location, day of week and season. This can effectively eliminate the effect of location and time on traffic status.

Due to the poor performance of detectors in real applications, the raw data contains some problems, such as missing data, noise data, etc. This study adopts techniques to preprocess raw data for better prediction performance (Ma et al., 2019b; Ma and Cheng, 2016). The preprocessing methods including missing-data deletion, missing-data imputation, data aggregation and normalization adopted in this study are introduced in Section 3.2.

2.2. Model development: LSTMDTR

LSTM is a state-of-the-art technique for time-series problems and is reported to obtain excellent performance in other research domains, such as speech recognition, human activity recognition. However, few studies applied LSTM to detect traffic crashes on freeways with traffic data. To explore the effectiveness of LSTM on crash detection, this study

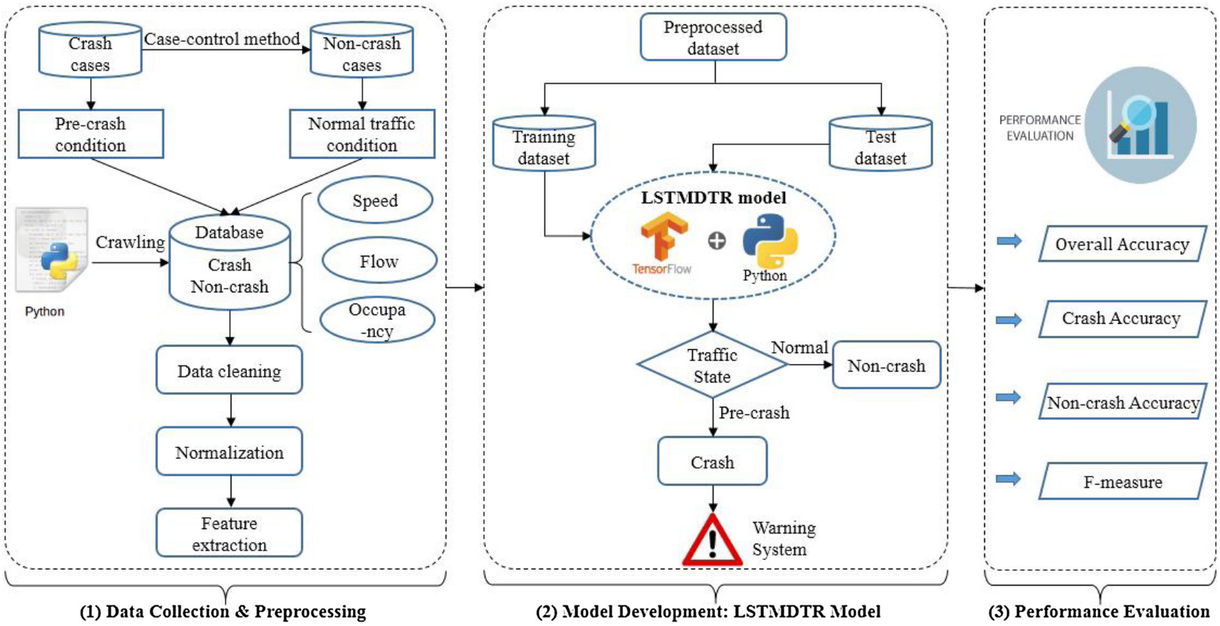

Fig. 1. Crash detection framework.

proposes an LSTMDTR model for crash detection, an LSTM-based framework with traffic data of different temporal resolutions. The novelty of the proposed model is that, unlike speech data or image data, this study designs three LSTM networks with different temporal resolutions to fit the specific property of traffic data. A fully-connected layer is used to combine the outputs of three LSTM networks, and a dropout layer is used to reduce overfitting and improve prediction performance.

Because of the specific property of traffic data, the LSTMDTR model has superiority on traffic crash detection:

Firstly, traffic data are time-series, which are commonly recorded every 30 s by loop detectors. Machine learning methods used in current studies cannot simulate the evolving transitions of traffic conditions and may impair the performance of crash detection (Hossain and Muromachi, 2012). However, LSTM can learn short-term and long-term dependencies in time-series data; therefore, capture the dynamic transition process of pre-crash conditions. This can help detect traffic crashes with better prediction performance.

Besides, traffic data of different temporal resolutions reflect different traffic conditions. The current studies established crash detection models with traffic data of only one temporal resolution, which cannot fully represent traffic variations in different periods (Sun and Sun, 2015). Therefore, this study combines three LSTM networks with traffic data of large, middle and small temporal resolutions. This can reflect traffic variations at different time intervals and improve detection performance.

2.2.1. LSTM network

LSTM network is a special deep learning method based on artificial recurrent neural network (RNN) for time-series issues (Ma et al., 2019a). As is shown in Fig. 2, unlike classical machine learning methods, both RNN and LSTM have feedback connections, which enable information transmission between time intervals (Ma et al., 2019d, 2019c). However, classical RNN encompasses exploding and vanishing gradient problems, which impair the ability to perceive information from distant intervals (Wunnava et al., 2018). Therefore, Hochreiter and Schmidhuber (1997) developed LSTM networks to address these limitations. LSTM network possesses the capacity to learn and model long-term dependencies in sequential data to improve prediction performance.

LSTM has a special memory mechanism to select and maintain information for a long period. As Fig. 2 shows, LSTM consists of multiple memory cells. Information is transmitted from the previous cell to the next one. Each cell receives input data of the current time step and information transmitted from the previous time steps. The cell contains three unique gates to control information transmission in long-term dependencies, namely forget gate, input gate, and output gate. The three gates are introduced as follows:

(1) The forget gate is designed to control the information from the previous cell that should be forgotten in the current cell. It receives input of the current cell x_t and output of the previous cell h_{t-1}, and returns an output of f_t ranging from 0 to 1 by a sigmoid function. The value of f_t indicates the extent of retained information in the current cell, where “0” donates “all forgotten” and “1” represents “all retained”. The equation is shown as follows:

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{1}$$

where f_t is the output of the forget gate; σ is the sigmoid function; W_f and b_f are weight and bias matrixes of the forget gate.

(2) The input gate is designed to determine the information of the current input that is permitted to flow into the current cell. This gate accepts input of the current cell x_t and output of the previous cell h_{t-1}, and then returns two parts of outputs: i_t by a sigmoid function and $\widetilde{C}_t$ by a tanh function. The equations are listed as follows:

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i \tag{2}$$

$$\widetilde{C}_t = tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{3}$$

where i_t and $\widetilde{C}_t$ are the outputs of the input gate; *tanh* is the tanh function; W_i and W_C are weight matrixes of the input gate; b_i and b_C are bias matrixes of the input gate; σ is interpreted in Eq. (1).As Fig. 2 shows, the cell state can be updated from C_{t-1} to C_t by integrating the outputs of forget gate and input gate. The current cell state C_t comprehensively considers the output of the forget gate f_t, two outputs of the input gate (i_t and $\widetilde{C}_t$), and the previous cell state C_{t-1}. The equation is written as follows:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{4}$$

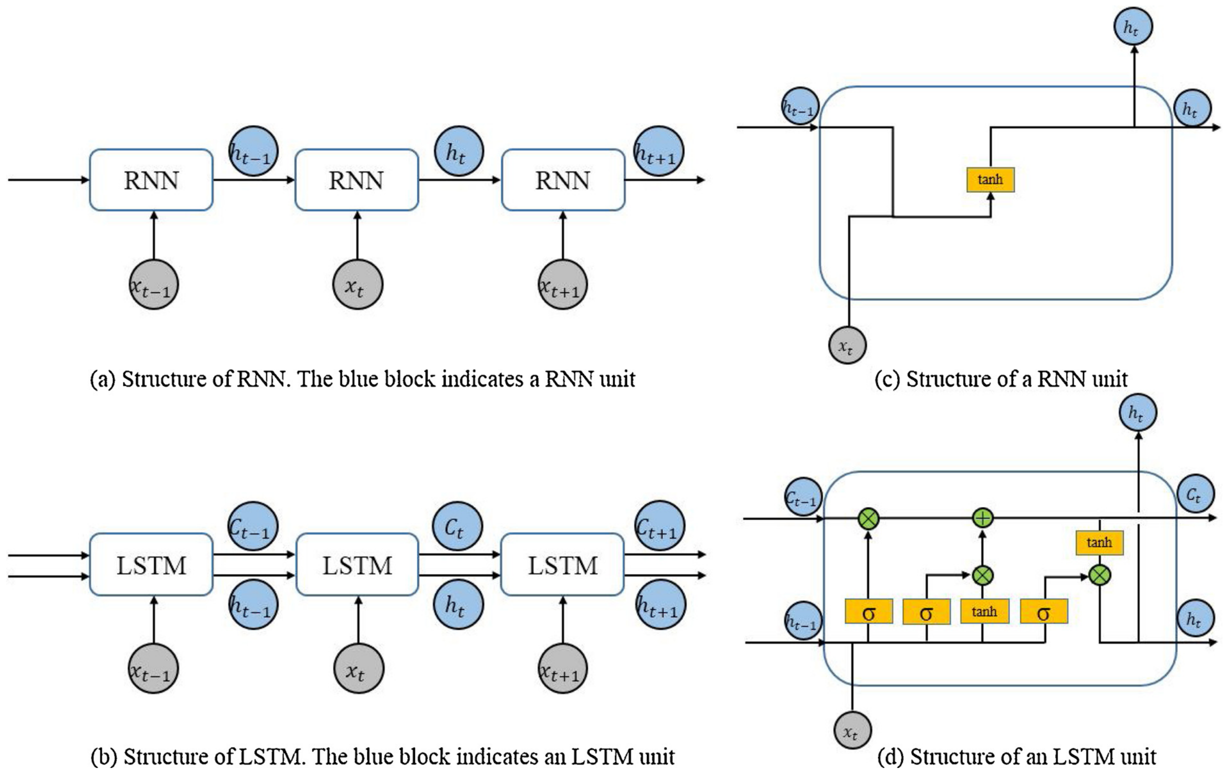

Fig. 2. Structure of RNN and LSTM.

where C_{t-1} and C_t are the previous and current cell state; f_t, i_t and $\widetilde{C}_t$ are interpreted in Eqs. (1)–(3).

(3) The output gate determines the extent to which information is outputted from the current cell. This gate receives input of the current cell x_t and output of the previous cell h_{t-1}, and returns an output of O_t by a sigmoid function. The calculation is shown in Eq. (5). However, the final output of the current cell h_t considers the output of the output gate O_t and the current cell state C_t, which is shown in Eq. (6).

$$O_t = \sigma(W_O \bullet [h_{t-1}, x_t] + b_O) \tag{5}$$

$$h_t = O_t * \tanh(C_t) \tag{6}$$

where O_t is the output of the output gate; h_t is the final output of the current cell; W_O and b_O are weight and bias matrixes of the output gate; σ, tanh and C_t are interpreted in Eqs. (1)–(4).

2.2.2. LSTMDTR model

To address the above research gaps and limitations in current studies, this study proposes an LSTMDTR model——LSTM based framework considering traffic data of different temporal resolutions for crash detection. The structure of the LSTMDTR model is shown in Fig. 3, including six kinks of layers, namely (1) input layer, (2) LSTM layer, (3) the first fully-connected layer, (4) dropout layer, (5) the second fully-connected layer, and (6) output layer.

This study aims to detect pre-crash conditions from normal traffic conditions. Both crash cases and non-crashes cases with corresponding traffic data are collected and preprocessed. This study selects traffic data of different temporal resolutions for the input layer, including large, middle, and small resolutions. This can comprehensively indicate traffic trends and variations in different time intervals to improve prediction performance. LSTM network is a powerful deep learning method to capture the long-term dependency and dynamic process of pre-crash conditions with time-series traffic data. Three LSTM networks are constructed to simulate and capture the dynamic transition process of pre-crash conditions within long, middle and short time intervals. The first fully-connected layer is used to integrate and flatten the outputs of three LSTM networks. A dropout layer is also used to reduce overfitting and improve the prediction performance of the LSTMDTR model. The neurons after dropout are calculated and transferred into the second fully-connected layer. And neurons in the second fully-connected layer are transferred to the output layer for binary classification, indicating "Crash" or "Non-crash".

The calculation process and relevant equations of the LSTMDTR model are shown as follows:

(1) Input layer: This study collects and preprocesses traffic data for the input layer. x^L, x^M and x^S represent preprocessed traffic data of large, middle, and small temporal resolutions.
(2) LSTM layer: Three LSTM networks with long, middle and short temporal resolutions are constructed to simulate the dynamic variations of traffic conditions in different time intervals. Eqs. (7)–(9) indicates the calculation of three LSTM networks.

$$h^L = LSTM(x^L) \tag{7}$$

$$h^M = LSTM(x^M) \tag{8}$$

$$h^S = LSTM(x^S) \tag{9}$$

where h^L, h^M and h^S are the outputs of three LSTM networks with traffic data of large, middle, and small temporal resolutions; x^L, x^M and x^S are preprocessed traffic data of large, middle, and small temporal resolutions; *LSTM* is the calculation process of LSTM networks, which is introduced in Section 2.2.1.

(3) The first fully-connected layer: The outputs of three LSTM networks are integrated and flattened by the first fully-connected layer. Eq. (10) indicates the calculation of the first fully connected layer.

$$H = F(h^L, h^M, h^S) \tag{10}$$

where H is the output of the first fully-connected layer; F is a fully-

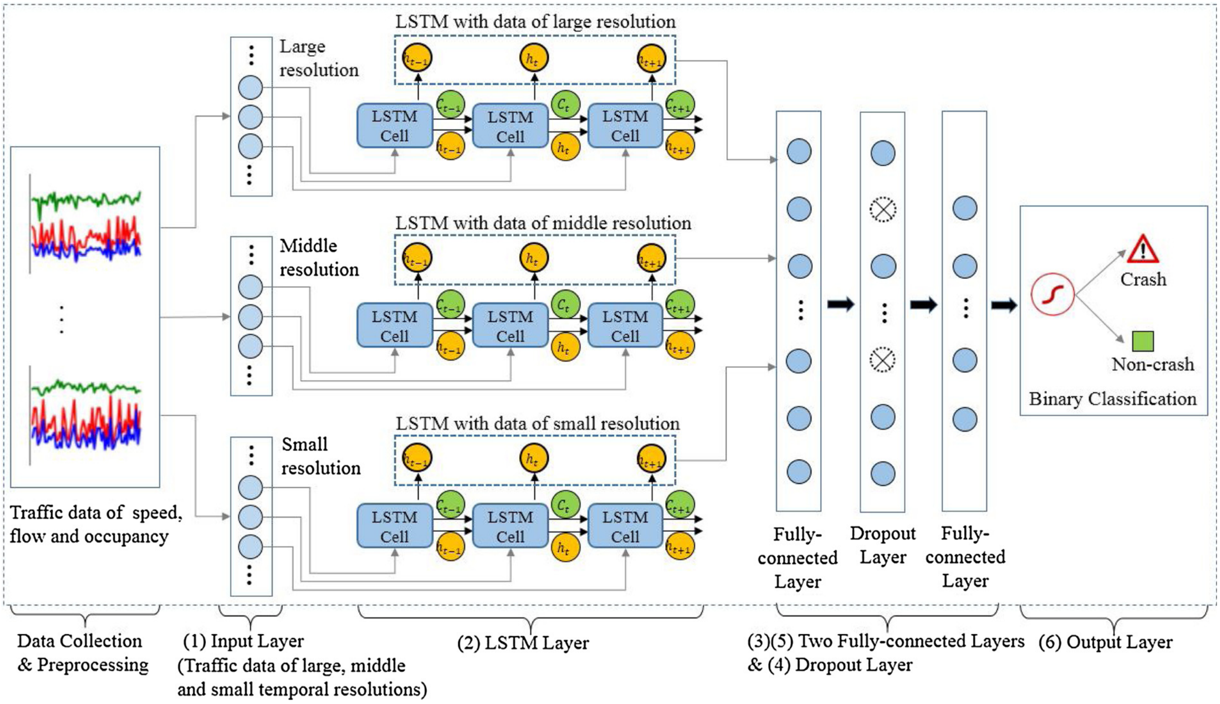

Fig. 3. Structure of LSTMDTR model.

connected layer to combine and flatten the outputs of three LSTM networks; h^L, h^M and h^S are introduced in Eqs. (7)–(9).

(4) Dropout layer: A dropout layer is adopted in this study to reduce overfitting and improve the prediction performance of the LSTMDTR model. The dropout technique randomly drops out hidden neurons from networks. Neurons can be more robust and insensitive by cooperating with a subset of randomly selected neurons. This can improve the generalization ability and prediction performance of deep learning networks on unseen data. Fig. 4 shows the structure and calculation process of a standard network and a dropout network. Eqs. (11)–(12) donates the calculation of the dropout layer.

$$r \sim Bernoulli(p) \tag{11}$$

$$\widetilde{H} = r * H \tag{12}$$

where r is a random variable donating by Bernoulli distribution, which takes 1 with probability p and 0 with probability $1 - p$; $\widetilde{H}$ donates randomly chosen neurons from H by dropout; H is explained in Eq. (10).

(5) The second fully-connected layer: The neurons after dropout are transferred and calculated in the second fully-connected layer. Fig. 4 shows the calculation process. Eq. (13) indicates the calculation of the second fully-connected layer.

$$S = f(w_s * \widetilde{H} + b_s) \tag{13}$$

where S is the output of the second fully-connected layer; f is a sigmoid function; w_s and b_s are weight and bias matrixes of the second fully-connected layer; $\widetilde{H}$ is explained in Eq. (12).

(6) Output layer: Neurons in the second fully-connected layer are transferred to the output layer for binary classification, indicating "Crash" or "Non-crash". Eq. (14) shows the calculation of the output layer.

$$O = f(w_o * S + b_o) \tag{14}$$

where O is the final output of binary classification, indicating "Crash" or "Non-crash"; w_o and b_o are weight and bias matrixes of the output layer; S and f are explained in Eq. (13).

2.3. Performance evaluation

Due to the imbalance of the dataset (i.e., more non-crash samples than crash samples), the indicator of overall accuracy cannot comprehensively evaluate the performance of the crash detection model. The reason is that overall accuracy is still very high, even though all the crashes are classified as non-crashes in an imbalanced dataset. Therefore, according to previous studies, four criteria are applied in this study to evaluate the performance of crash detection model: overall accuracy, true positive rate (crash accuracy), true negative rate (non-crash accuracy) and false alarm rate (FAR) (Sun et al., 2014). They are constructed based on the confusion matrix in Table 1.

Overall accuracy is defined as the ratio of cases that are correctly classified to the total number of cases. The equation is shown as follows:

$$Overall\ Accuracy = \frac{T_{Crash} + T_{Non-crash}}{T_{Crash} + F_{Crash} + F_{Non-crash} + T_{Non-crash}} * 100\% \tag{15}$$

True positive rate and true negative rate are used to indicate the prediction performance of positive class and negative class. In this study, the true positive rate is defined as crash accuracy, which is the proportion of correctly classified crash cases to the total number of crash cases. The true negative rate is defined as non-crash accuracy, which is the proportion of detected non-crash cases to the total number of non-crash cases. The equations are shown as follows:

$$Crash\ Accuracy = \frac{T_{Crash}}{T_{Crash} + F_{Non-crash}} * 100\% \tag{16}$$

$$Non-Crash\ Accuracy = \frac{T_{Non-crash}}{F_{Crash} + T_{Non-crash}} * 100\% \tag{17}$$

False alarm rate (FAR) is the proportion of false alarm cases to the total number of non-crash cases. The equation is written as follows:

$$\text{False Alarm Rate(FAR)} = \frac{F_{Crash}}{F_{Crash} + T_{Non-crash}} * 100\% \tag{18}$$

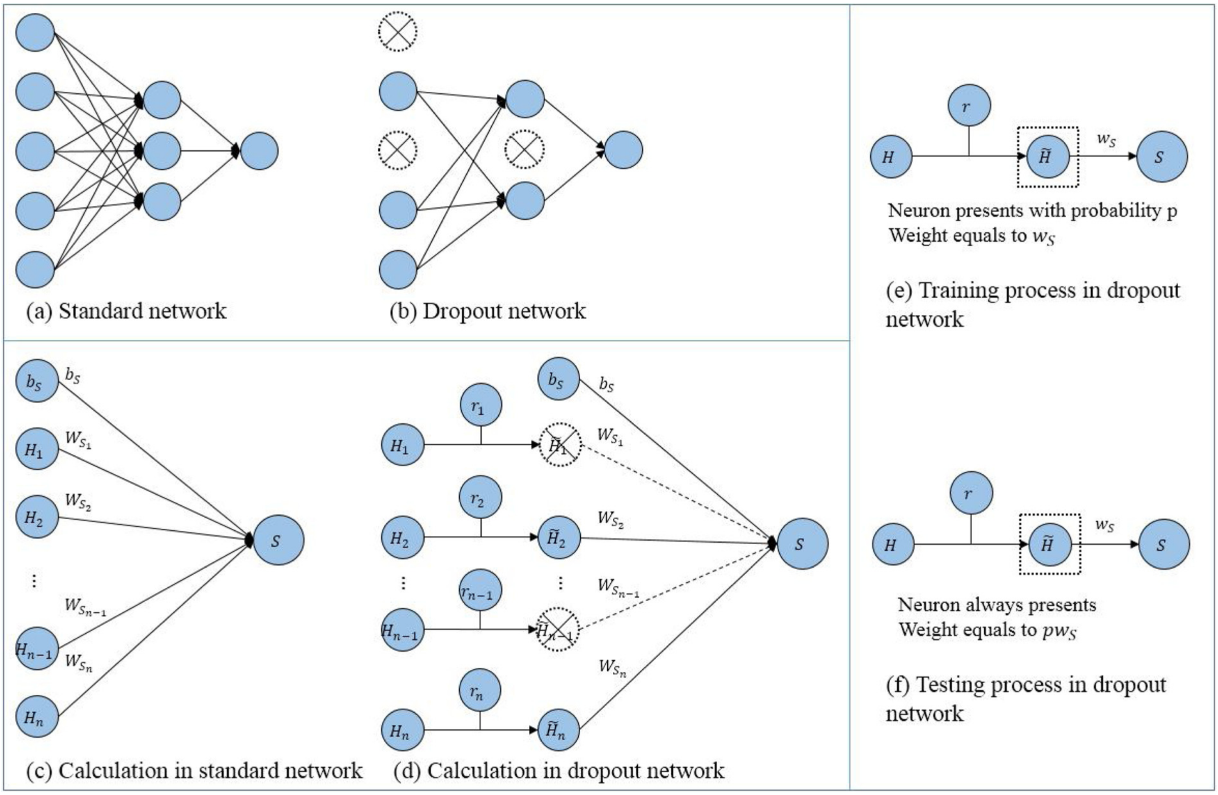

Fig. 4. Structure and calculation process of standard network and dropout network.

Table 1
Confusion matrix.

	Predicted: Crash	Predicted: Non-crash
Observed: Crash	T_{Crash}	$F_{Non-crash}$
Observed: Non-crash	F_{Crash}	$T_{Non-crash}$

3. Case study

3.1. Study area

This study proposes an LSTMDTR model for crash detection with traffic data on freeways. The detection model aims to explore the inner relationship between traffic conditions and crash risk (Hossain and Muromachi, 2012). According to previous studies, there exist two kinds of traffic conditions (Sun and Sun, 2015). Traffic condition before an accident is defined as a pre-crash condition because it is closely associated with the crash occurrence. And traffic condition under specific criteria that does not lead to an accident is defined as normal traffic condition. The detection model is designed to classify pre-crash conditions from normal traffic conditions. Therefore, both crash samples (i.e., pre-crash conditions) and non-crash samples (i.e., normal traffic conditions) with relevant traffic data should be collected for crash detection on freeways.

This study collects data from I880-N and I805-N in California, America. Both these two roads are very important interstate highways in California with intelligent sensing networks. I880-N is about 46 miles long with 203 loop stations and 592 detectors, while I805-N is about 28.7 miles long with 124 loop stations and 333 detectors. The loop stations spread with an interval of about 0.3 miles. The detectors collect and record data of speed, flow, and occupancy in each lane every 30 s. The data collected from I880-N is used to evaluate the LSTMDTR model on crash detection. And the data collected from I805-N is used to test model transferability on similar freeways. The data is collected from the Freeway Performance Measurement System (PeMS) in California.

3.2. Data collection and preprocessing

This study aims to use traffic conditions to detect traffic accidents. Therefore, both crash cases and non-crash cases with corresponding traffic data should be collected and preprocessed. This study collects crash cases of I880-N between January 2017 to December 2018 and crash cases of I805-N in January 2018. Crash cases contain information of crash date and time, duration, location, accident severity, etc. The non-crash cases are collected at the same time, location, day of week and season of crash cases. This can effectively eliminate the effect of location and time on traffic conditions. This study collects traffic data of 35 min. before a crash /non-crash (Hossain and Muromachi, 2012; Sun and Sun, 2015). The traffic data contains speed, flow, and occupancy every 30 s on each lane. They are collected from 6 detector stations nearest to crash/non-crash locations, with 3 stations upstream and 3 stations downstream, which is shown in Fig. 5. Traffic data of prior 30 min. are used to predict if crashes will occur within the next five minutes.

It is worth mentioning that this study develops a Python tool package to download traffic data automatically. Fig. 6 shows the diagram of crawling webpage for data collection. The URLs are constructed by the information of crash time and location. The package can automatically download traffic data by fetching URLs. This tool is designed to collect traffic data for both crash cases and non-crash cases.

However, there exist many missing values in the collected raw data. If the missing rate of a sample is very high, it may bring in different levels of noise to the experiments. Existing literature often discarded samples with many missing data (Motamed, 2016; Xu et al., 2014). This study also deletes the samples with a high missing rate. Besides, there are other preprocessing steps undertaken in this study. Details are as follows:

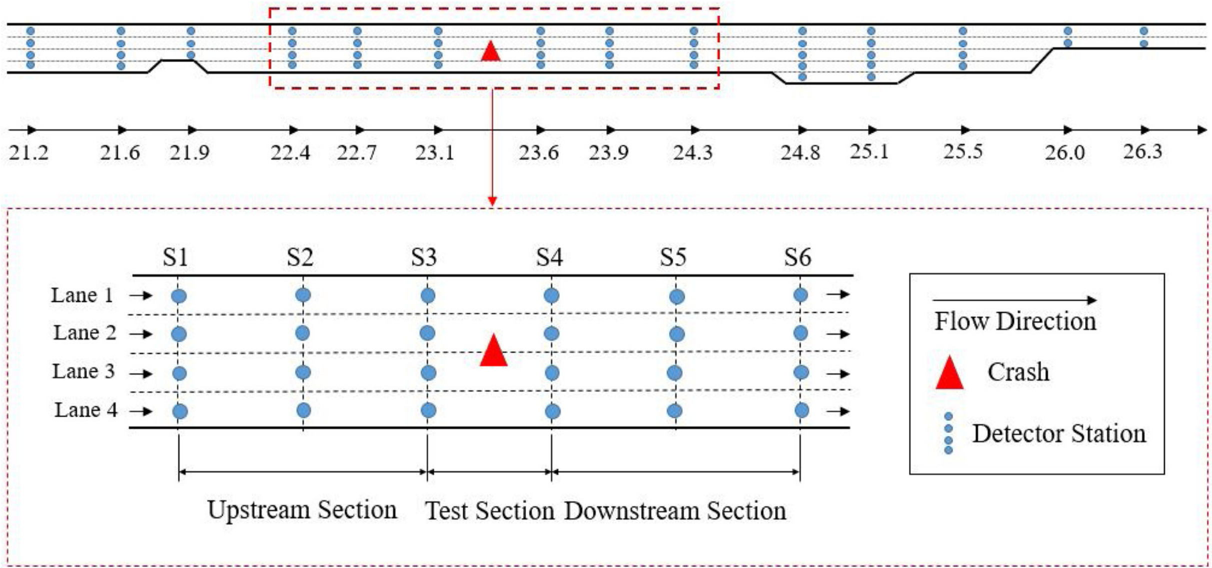

Fig. 5. Diagram of traffic data collection.

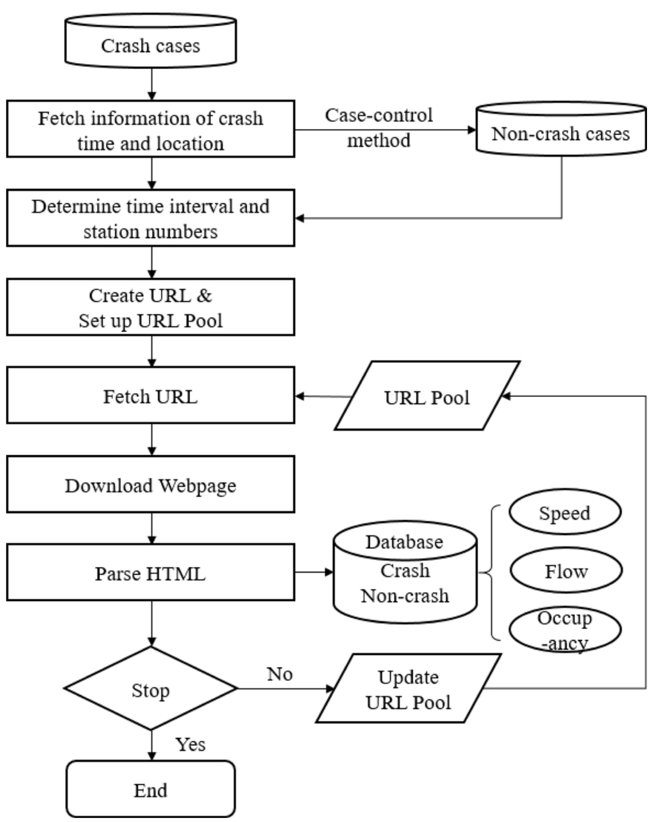

Fig. 6. Diagram of crawling webpage for data collection.

- Missing data: This study implements deletion and imputation to deal with missing data. These include: (1) The cases with no data recorded at more than one detector stations are deleted; (2) The cases with more than 20 % of missing data are deleted; (3) The cases with missing data less than 20 % are imputed with the value of last moment.
- Data aggregation: Traffic data in this study is collected every 30 s in each lane of each station. Since one station may have multiple lanes (same direction), to reduce computational complexity and support real-time detection, this study adopts the following steps to aggregate traffic data for each station: (1) Average the traffic data of all the lanes to represent the traffic condition of a station; (2) The averaged traffic data of a station is aggregated into different time intervals (i.e., 1 min, 5 min., and 10 min.), which can reduce computational complexity and fully represent traffic variations in different periods.
- Data normalization: This study adopts z-score to normalize traffic data. Z-score standardizes features by removing the mean and scaling to unit-variance (Cheng and Ma, 2015a, 2015b). This can reduce the risk that the results are governed by some extreme values.

After data preprocessing, the dataset of I880-N between January 2017 to December 2018 is transformed into 1386 crash samples and 2781 non-crash samples. The dataset of I805-N in January 2018 contains 43 crash cases and 89 non-crash cases. These datasets are used to construct crash detection models to validate the assumption that crashes can be predicted by traffic flows.

4. Results

This study applies the proposed LSTMDTR model on case studies for crash detection with traffic data of different temporal resolutions. The dataset of I880-N is used to examine the effectiveness of the LSTMDTR model on crash detection, with 1386 crash cases and 2781 non-crash cases between January 2017 to December 2018. The dataset of I805-N is used to test model transferability on similar freeways, with 43 crash cases and 89 non-crash cases in January 2018.

This study randomly selects 80 % of cases for training, 10 % of cases for validation, and 10 % of cases for testing. To reduce randomness and improve the reliability of the detection model, this study conducts experiments for five times and averages the criteria. The experiments are implemented by Python code and a Tensorflow-GPU platform.

4.1. Determination of temporal resolutions and parameters in LSTMDTR model

4.1.1. Determination of temporal resolutions

The current studies used traffic data of only one temporal resolution, which cannot fully represent traffic variations in different time intervals and may impair the model performance on crash detection. Therefore, this study proposes an LSTMDTR model considering traffic

Table 2
Definitions for temporal resolutions.

Temporal Resolution	Reaction Time	Time Interval	Aggregation Interval
Large	0-5 min.	5–35 min.	10 min.
Middle		5–20 min.	5 min.
Small		5–10 min.	1 min

data of different temporal resolutions to improve prediction performance. It is important to determine the reaction time and temporal resolutions of traffic data in the LSTMDTR model.

Almost all the studies defined reaction time as 0–5 min. before crash occurrence, providing enough buffer time to implement prevention measures (Sun et al., 2014; Xiao, 2019). This study also determined the last 5 min. before a crash as reaction time.

The current studies used traffic data within 30 min. for crash detection. This is because traffic data larger than 30 min. (e.g., 1 h) has less effect on crash occurrence (Hossain and Muromachi, 2012; Liu and Chen, 2017). For example, Liu and Chen (2017) validated that traffic data of 30 min. before crash occurrence can obtain the best detection performance. Sun and Sun (2015) used traffic data of 5–20 min. before crash occurrence for crash detection. Sun et al. (2014) developed models with traffic data of 5–10 min. before a crash. The current studies commonly used large aggregation intervals for large time spans and small aggregation intervals for small time spans (Hossain et al., 2019).

According to the previous studies, this study defines three temporal resolutions in Table 2, namely large, middle, and small resolutions. The time intervals for large, middle and small resolutions are 5–35, 5–20 and 5–10 min. before crash occurrence with traffic data aggregated into 10, 5 and 1 min. Traffic data of large temporal resolution reflect traffic trends within a long period and traffic data of small temporal resolutions reflect traffic variations within a short period. The three temporal resolutions can fully represent traffic variations in different time intervals and improve model performance on crash detection.

4.1.2. Determination of parameters in LSTMDTR model

The parameters have an impact on the results of the LSTMDTR model on crash detection. However, it is time-consuming to optimize all the parameters to obtain the best results. Therefore, this study determines the important parameters according to the previous studies (Hochreiter and Schmidhuber, 1997; Lei et al., 2019; Wang et al., 2019). Important parameters and their settings are listed as follows: the number of neurons in LSTM is 64; the dropout rate is 0.4; the number of neurons in the second fully-connected layer is 64; the epoch is 100; the batch size is 50; the optimization algorithm is Adam.

4.2. Results of LSTMDTR on crash detection

As is shown in Table 3, this study establishes 6 LSTMDTR models with different features on crash detection. For example, traffic features for $M_{S_6{*}F_{SFO}}$ model contain data of speed, flow, and occupancy from 6 nearest stations (3 stations upstream and 3 stations downstream). Traffic features for $M_{S_4{*}F_{SFO}}$ model contain data of speed, flow, and occupancy from 4 nearest stations (2 stations upstream and 2 stations downstream). As speed condition has a dominant effect on crash detection in previous research, this study also constructs LSTMDTR models only with speed data (Hossain and Muromachi, 2012; Sun and Sun, 2015). For example, traffic features for $M_{S_4{*}F_S}$ model are speed data from 4 nearest stations (2 station upstream and 2 station downstream). Traffic features for $M_{S_2{*}F_S}$ model are speed data from 2 nearest stations (1 station upstream and 1 station downstream). It is worth mentioning that this study collects traffic data of 2, 4 and 6 nearest stations because almost all the current studies uses traffic data of 2, 4 or 6 stations for crash detection (Abdel-Aty and Pande, 2005; Hossain and Muromachi, 2012; Xu et al., 2013). The results of LSTMDTR models on crash prediction are shown in Table 4.

Table 3
Description of LSTMDTR models.

Model	Features	Stations	Inputs for Each LSTM	Inputs for LSTMDTR
$M_{S_6{*}F_{SFO}}$	Speed, Flow, Occupancy	6	Large: 3*6*3 = 54 Middle: 3*6*3 = 54 Small: 3*6*5 = 90	198
$M_{S_6{*}F_S}$	Speed	6	Large: 6*3 = 18 Middle: 6*3 = 18 Small: 6*5 = 30	66
$M_{S_4{*}F_{SFO}}$	Speed, Flow, Occupancy	4	Large: 3*4*3 = 36 Middle: 3*4*3 = 36 Small: 3*4*5 = 60	132
$M_{S_4{*}F_S}$	Speed	4	Large: 4*3 = 12 Middle: 4*3 = 12 Small: 4*5 = 20	44
$M_{S_2{*}F_{SFO}}$	Speed, Flow, Occupancy	2	Large: 3*2*3 = 18 Middle: 3*2*3 = 18 Small: 3*2*5 = 30	66
$M_{S_2{*}F_S}$	Speed	2	Large: 2*3 = 6 Middle: 2*3 = 6 Small: 2*5 = 10	22

Table 4
Results of LSTMDTR models on crash detection.

Model	Overall Accuracy (%)	Crash Accuracy (%)	Non-Crash Accuracy (%)	FAR (%)
$M_{S_6{*}F_{SFO}}$	78.75	70.43	82.89	17.11
$M_{S_6{*}F_S}$	82.31	64.35	91.26	8.74
$M_{S_4{*}F_{SFO}}$	78.31	65.22	84.84	15.16
$M_{S_4{*}F_S}$	85.54	63.33	96.61	3.39
$M_{S_2{*}F_{SFO}}$	82.89	61.88	93.36	6.64
$M_{S_2{*}F_S}$	85.69	59.28	98.84	1.16

The results indicate that the proposed LSTMDTR models can obtain satisfactory performance on crash detection. All the models can predict crashes and non-crashes with an overall accuracy higher than 75 %. Most models can detect crashes with crash accuracy higher than 60 % and non-crash accuracy higher than 80 %. Such prediction performance is acceptable in the domain of crash detection with traffic data.

Crash accuracy is the most important evaluation criterion for crash detection. $M_{S_6{*}F_{SFO}}$ model can obtain the highest crash accuracy of 70.43 %. This crash accuracy is higher than many previous studies on crash prediction. For example, 59.64 % of crash accuracy can be obtained using an SVM and KNN ensemble model by Xiao (2019). 61 % of crashes can be detected using RF and genetic programming model by Xu et al. (2013). 66 % of crash accuracy can be achieved using a Bayesian network by Hossain and Muromachi (2012). This indicates that the proposed LSTMDTR model can achieve better performance on crash prediction.

Table 4 shows that non-crash accuracy is higher than crash accuracy in LSTMDTR models. This may be caused by the imbalanced dataset, with more non-crash cases than crash cases. More non-crash cases are trained by the detection model; therefore, the model tends to have better performance of non-crash accuracy. However, this study cannot simply rely on the overall accuracy or the non-crash accuracy to select the best model. Instead, this study should focus more on crash accuracy, which is also called the “true positive rate” measurement in data mining. This is because models with higher crash accuracy can predict more potential crashes and implement measures to prevent crashes in advance. Therefore, $M_{S_6{*}F_{SFO}}$ model is selected as the best crash detection model in this study. It has the highest crash accuracy even though the lowest non-crash accuracy.

To realize real-time prediction, it is important to analyze the computational time of detection models. The proposed $M_{S_6{}^*F_{SFO}}$ model is selected as the best model for crash detection with traffic data of speed, flow, and occupancy from 6 nearest stations. The experiments are conducted in a computer with an i5-6500 CPU at 3.20 GHz and Windows 10 of 64-bit OS. The testing time for each sample is 0.92 s in this study, including data preparation and testing. This running time is much shorter than 5 min., which is the time span of real-time traffic data collection. Therefore, the $M_{S_6{}^*F_{SFO}}$ models can support real-time crash detection in practical applications.

Base on the results in Table 4, two things are worth mentioning in this study. Firstly, models with data of speed, flow, and occupancy can obtain higher crash accuracy than those with only speed data. Crash accuracy can be increased by 2 %–6 % due to flow and occupancy data. This is because flow and occupancy data can provide more information for LSTMDTR models to learn pre-crash conditions and improve detection performance. Secondly, models with data from 6 stations have better prediction performance than 4 or 2 stations. Especially, models with data from 2 stations can hardly detect crashes. This may be that traffic information from 2 stations is limited. They cannot represent the real traffic state and may impair the detection performance of the LSTMDTR model.

4.3. Results of LSTMDTR on transferability

This study conducts experiments to test the spatial transferability of LSTMDTR models on crash detection. Transferability aims to evaluate if detection models established on one freeway can be transferred to predict crashes on another similar freeway. As is shown in Table 5, this study uses the dataset of I880-N to train LSTMDTR models and uses the dataset of I805-N to test the performance of model transferability. 6 LSTMDTR models with different features are constructed in this study. The results of model transferability are shown in Table 6.

Compared with the models established on the same freeway, the transferability models have a decrease in crash detection, especially on the criterion of crash accuracy (Tables 4 and 6). This may be that traffic conditions, road geometry, vehicle types, and accident patterns in different freeways are different. Therefore, the transportation authorities need to select freeways as similar as possible when applying transferability models on crash detection.

However, Table 6 indicates that some transferability models can predict crashes with acceptable performance. For example, transferability models with data of speed, flow, and occupancy from 6 or 4 stations (i.e., $M_{S_6{}^*F_{SFO}}$ and $M_{S_4{}^*F_{SFO}}$) can detect crashes with crash accuracy higher than 59 %. Especially, $M_{S_6{}^*F_{SFO}}$ model can predict 65.12 % of crashes with 72.25 % of overall accuracy. This crash accuracy of model transferability is higher than those in previous studies. For example, Pande et al. (2012) applied an LR model on transferability with the highest crash accuracy of 55.84 %. Xu et al. (2014) used the Bayesian updating approach on transferability with 59.3 % of crash accuracy. Therefore, the proposed LSTMDTR models constructed on one freeway can be transferred to other similar freeways for crash detection if proper features and networks are provided.

Table 6
Results of LSTMDTR models on transferability.

Model	Overall Accuracy (%)	Crash Accuracy (%)	Non-Crash Accuracy (%)	FAR (%)
$M_{S_6{}^*F_{SFO}}$	72.25	65.12	75.81	24.19
$M_{S_6{}^*F_S}$	77.83	50.70	91.40	8.60
$M_{S_4{}^*F_{SFO}}$	65.58	59.07	68.84	31.16
$M_{S_4{}^*F_S}$	81.09	49.77	96.74	3.26
$M_{S_2{}^*F_{SFO}}$	78.76	50.23	93.02	6.98
$M_{S_2{}^*F_S}$	82.02	46.98	99.53	0.47

5. Discussions

5.1. Comparison with machine learning methods on crash detection and transferability

To validate the effectiveness of the proposed methodology, this study compares the results of LSTMDTR model with some machine learnings, such as artificial neural networks (ANN), naive Bayes (NB), logistic regression (LR), support vector machine (SVM), k-nearest neighbor (KNN), and random forest (RF). These machine learning methods are extensively applied on crash detection in current studies and selected for comparison in this study (Abdel-Aty et al., 2012; Ahmed and Abdel-Aty, 2012; Liu et al., 2014; Xiao, 2019; Yao et al., 2014; Yuan and Cheu, 2003). The inputs for these machine learning models are speed, flow, and occupancy data with three different temporal resolutions (i.e., large, middle and small) from 6 nearest stations (3 stations upstream and 3 stations downstream).

It is worth mentioning that crash accuracy is the most important criterion for performance evaluation on an imbalanced dataset. Higher crash accuracy means more potential crashes can be detected in advance and effective measures can be implemented to improve road safety, which is the main purpose of this study.

Fig. 7 shows the comparison results on crash detection based on the dataset of I880-N. The results indicate that the proposed LSTMDTR model has better prediction performance on crash detection, especially on the criterion of crash accuracy. LSTMDTR model can detect 70.43 % of crashes with 78.75 % of overall accuracy, while crash accuracies of other machine learning methods are all lower than 65 %.

Fig. 8 shows the comparison results on model transferability based on the datasets of I880-N and I805-N. The results indicate that the proposed LSTMDTR model has better performance on transferability than other machine learning methods according to the criterion of crash accuracy. LSTMDTR model can predict 65.12 % of crashes with 72.25 % of overall accuracy, while crash accuracies of other machine learning methods are all lower than 60 %. This indicates that the LSTMDTR model established on one freeway can be transferred to other similar freeways for crash detection.

Based on Figs. 7 and 8, it is noteworthy that both KNN and LG are not effective methods for crash detection and model transferability. This result is consistent with the previous studies (Sun et al., 2014). One possible explanation is that KNN and LG have defects in complicated non-linear problems. Other machine learning methods, such as ANN, SVM, and RF are effective methods for non-linear problems and have better performance than KNN and LG. However, the proposed deep learning method LSTMDTR can obtain the highest crash accuracy on crash detection and model transferability. Few studies used deep

Table 5
Details of datasets on transferability.

Dataset	Freeway	Time Range	Crash Cases	Non-Crash Cases
Training data	I880-N, California	2017.01-2018.12	1386	2781
Test data	I805-N, California	2018.01	43	89

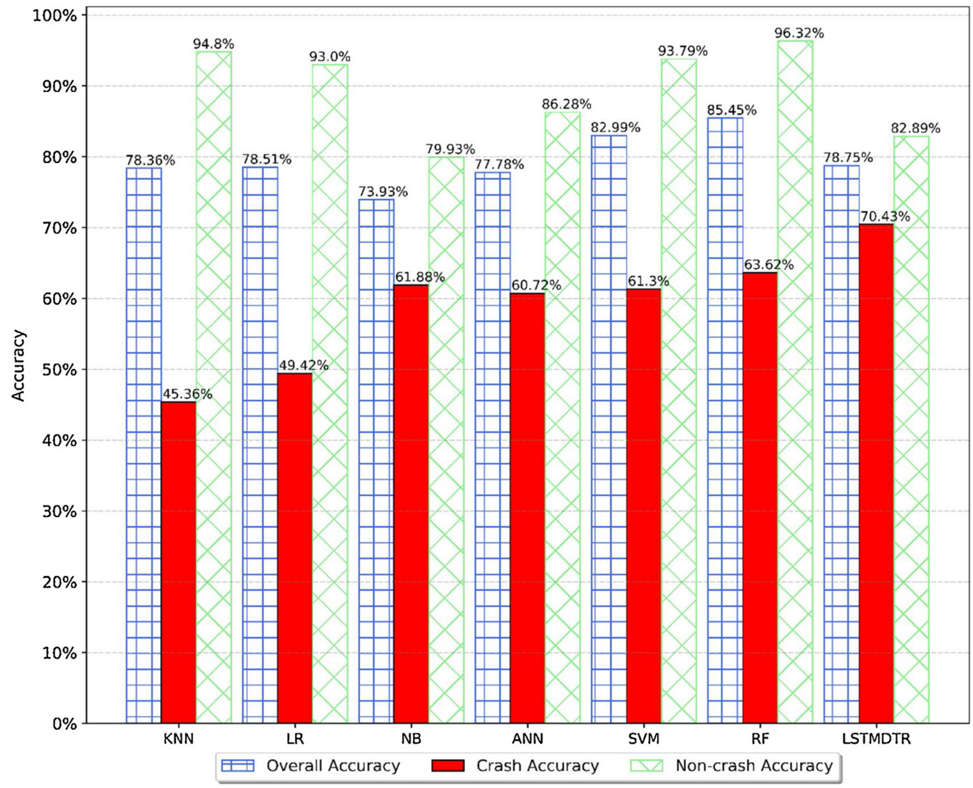

Fig. 7. Comparison with machine learning methods on crash detection.

learning methods for crash detection. This indicates that the importance of applying deep learning methods on crash detection was overlooked in the previous studies.

According to the above discussions, the LSTMDTR model is better than machine learning methods on crash detection. One potential reason is that crashes are caused by the disturbance of traffic conditions. It is necessary to capture the evolving process of pre-crash conditions by considering time-series traffic data. However, the classical

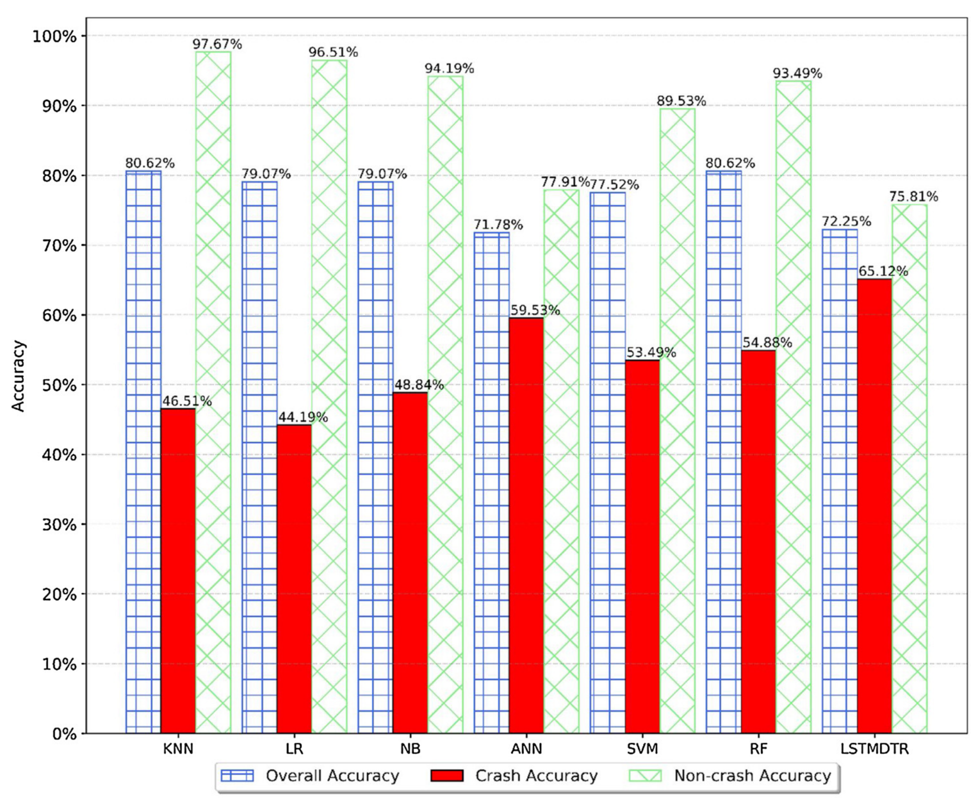

Fig. 8. Comparison with machine learning methods on transferability.

machine learning methods with flatted structures cannot simulate the dynamic variations of traffic conditions; therefore, impair the performance on crash detection. The proposed LSTMDTR model is established on LSTM, which is an effective deep learning method for time-series problems. LSTM has feedback connections and a specific mechanism to illustrate long-term dependencies in sequential data. This enables information transmission between time intervals and improves prediction performance. Therefore, the proposed LSTMDTR model can capture the dynamic transitions of pre-crash conditions and detect crashes with better performance.

5.2. The benefit of using three different temporal resolutions

The raw traffic data are collected every 30 s, containing speed, flow and occupancy data from 6 nearest stations. All the traffic data within 30 min. are very large, containing 60*3*6 dimensions. Detection models established on such a large size of inputs will have great training difficulty and calculational complexity. This may impair real-time applications of crash detection models. Therefore, this study reconstructs the raw data into three sets of temporal resolutions to train the model (i.e., large, middle, and small). Traffic data of large temporal resolution reflects traffic trends within a long period and traffic data of small temporal resolutions represent traffic variations within a short period. The three temporal resolutions in the LSTMDTR model can comprehensively indicate traffic trends in different time intervals; therefore, improve model performance on crash detection.

To validate the effectiveness of three temporal resolutions, this study compares the results of the proposed LSTMDTR model with those of one or two temporal resolutions. The comparison models are established on LSTM networks. The traffic data contains speed, flow and occupancy data from 6 nearest stations. Table 7 shows the description of comparison models. Fig. 9 shows the comparison results on crash detection, and Fig. 10 shows the comparison results on transferability.

Figs. 9 and 10 indicate that the proposed LSTMDTR model performs better than comparison models with one or two temporal resolutions. LSTMDTR model can obtain 70.43 % of crash accuracy on crash detection, while crash accuracies of comparison models are all lower than 69 %. LSTMDTR model can obtain 65.12 % of crash accuracy on transferability, while crash accuracies of comparison models are all lower than 62 %. Crash accuracy is the most important evaluation criterion for crash detection because effective measures can be implemented to prevent potential crashes and improve road safety. Therefore, the results indicate that the LSTMDTR model considering traffic data of three temporal resolutions has better performance on crash detection than those with one or two temporal resolutions.

LSTMDTR model can detect crashes with higher crash accuracy. This is because the LSTMDTR model encompasses three LSTM networks, which consider traffic data of large, middle, and small temporal resolutions. This can fully capture dynamic variations of traffic states at different time intervals and improve prediction performance. Almost all the previous studies used traffic data of only one temporal resolution. This indicates that the importance of considering traffic data of different temporal resolutions was overlooked in the previous studies.

Table 7
Description of models: LSTMDTR model with three temporal resolutions and comparison models with one or two temporal resolutions.

Number of Resolutions	Model	Temporal Resolution	Time Period	Aggregation Interval	Inputs for Model
1	LSTM(L)	Large	5-35 min.	10 min.	54
1	LSTM(M)	Middle	5-20 min.	5 min.	54
1	LSTM(S)	Small	5-10 min.	1 min	90
2	LSTM(LM)	Large, Middle	5-35 min., 5-20 min.	10 min., 5 min.	108
2	LSTM(LS)	Large, Small	5-35 min., 5-10 min.	10 min., 1 min	144
2	LSTM(MS)	Middle, Small	5-20 min., 5-10 min.	5 min., 1 min	144
3	LSTMDTR	Large, Middle, Small	5-35 min., 5-20 min., 5-10 min.	10 min., 5 min., 1 min	198

5.3. Effect of important parameters in LSTMDTR model

5.3.1. Effect of neuron numbers on crash detection

The number of neurons in hidden layers is an important parameter in the LSTMDTR model. It can simultaneously affect prediction performance and computation time on crash detection. Fig. 11 shows the results of $M_{S_6 * F_{SFO}}$ model with different numbers of neurons.

The results indicate that the number of neurons can affect prediction performance in LSTMDTR models. If proper numbers of neurons are provided (e.g., 32, 64), the crash accuracy can be very high, and the difference of crash accuracy is within 2 %, not to a large extent. However, the number of neurons cannot be too small or too large (e.g., 4, 8, 16, 128), because this may lead to underfitting (e.g., 4, 8, 16) or overfitting (e.g., 128) of LSTMDTR models and significantly affect the performance of crash detection.

Fig. 11 also shows that computation time increases by increasing the number of neurons in LSTMDTR models. Computation time is also an important consideration in real applications because models with less computation time can provide more time to take measures to prevent crashes. Therefore, a proper number of neurons should be determined in real applications considering acceptable detection performance and computation time.

5.3.2. Effect of dropout technique on crash detection

To prevent overfitting in complicated networks, this study adds a dropout layer in the LSTMDTR model to improve generalization ability on crash detection. The parameter of the dropout rate refers to the percentage of neurons that will be randomly excluded in the training process. According to previous studies, the dropout rate is selected between 0–70 %, where 0 represents no dropout is conducted in the model (Srivastava et al., 2014). The dropout rate cannot be too large (e.g., 80 %, 90 %) because this may cause the underfitting of LSTMDTR models. Fig. 12 shows the results of $M_{S_6 * F_{SFO}}$ model with different dropout rates.

The results indicate that the dropout method can improve prediction performance for LSTMDTR models. As Fig. 12 shows, without dropout technique (i.e., dropout rate = 0), crash accuracy can only be 64.49 %. However, with the effect of randomly eliminating neurons from networks, crash accuracy can be increased and ranges between 66.38 % to 70.43 %. One potential reason is that hidden neurons with a dropout layer should learn to be independent without relying on other neurons to correct their mistakes (Srivastava et al., 2014). This can improve the generalization ability of networks and obtain desirable performance even on unseen data. Therefore, the dropout technique can reduce overfitting and improve prediction performance in complicated networks.

Fig. 12 also shows that prediction performance fluctuates with the dropout rate. Crash accuracy increases when the dropout rate ranges from 10 % to 40 %. The model can obtain the highest crash accuracy of 70.43 % when the dropout rate is 40 %. However, when the dropout rate ranges from 50 % to 70 %, crash accuracy experiences a decrease. This may be that too many neurons are eliminated from networks and cause underfitting of the LSTMDTR model. Therefore, it is important to optimize the dropout rate in real applications to obtain optimal performance of crash detection.

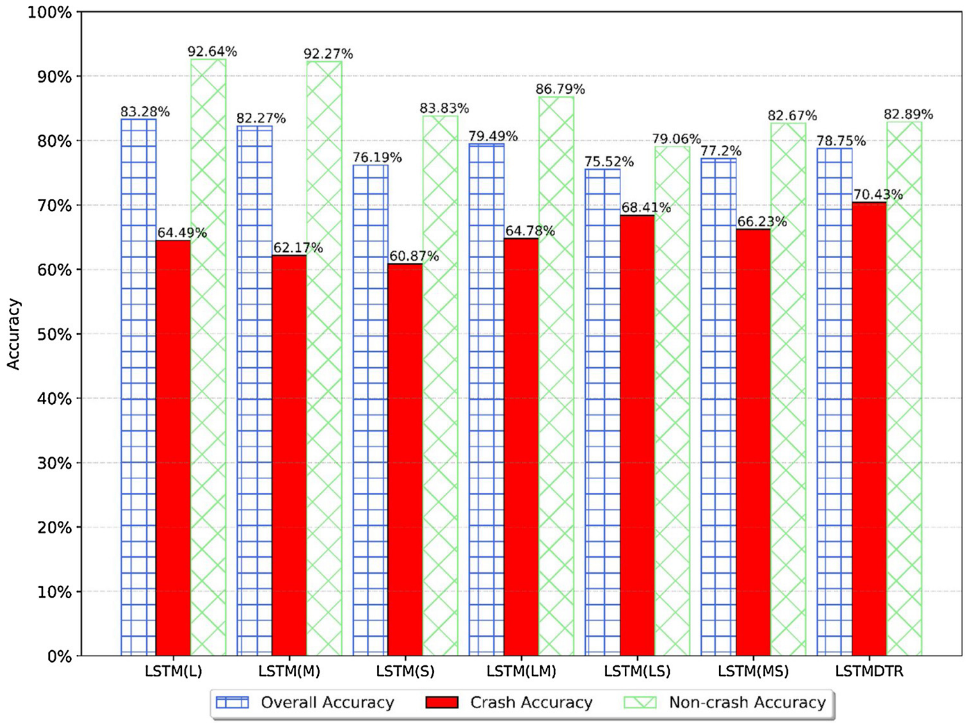

Fig. 9. Comparison results of models with different temporal resolutions on crash detection.

6. Conclusions

This study proposes an LSTM-based framework considering traffic data of different temporal resolutions (LSTMDTR) for crash detection on freeways. LSTM network is adopted in this study to capture the long-term dependency and dynamic variations of pre-crash conditions. Three LSTM networks considering traffic data of different temporal resolutions are constructed, which can comprehensively indicate traffic trends and variations in different time spans. A fully-connected layer is used to combine the outputs of three LSTM networks, and a dropout layer is used to reduce overfitting and improve prediction performance. The LSTMDTR model is implemented on datasets of I880-N and I805-N in California, America. The findings and contributions are summarized as follows:

(1) This study is one of the few papers using deep learning methods on crash detection, and the first one to consider traffic data of different temporal resolutions. The results indicate that the proposed

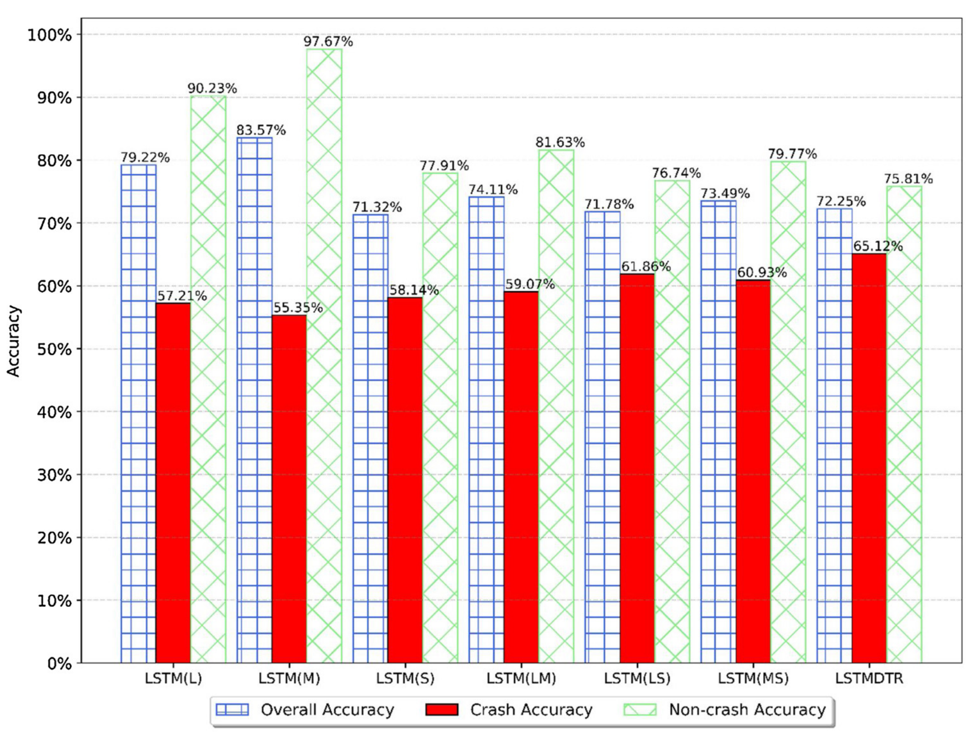

Fig. 10. Comparison results of models with different temporal resolutions on transferability.

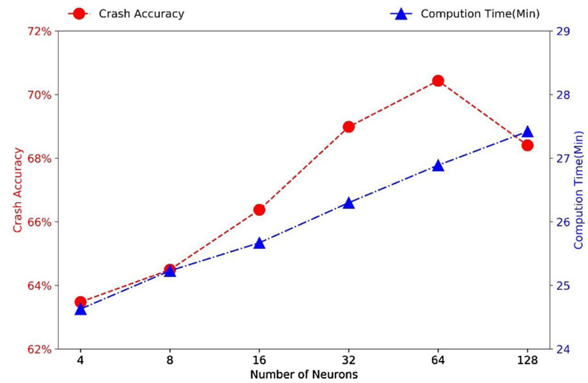

Fig. 11. Results of model with different numbers of neurons.

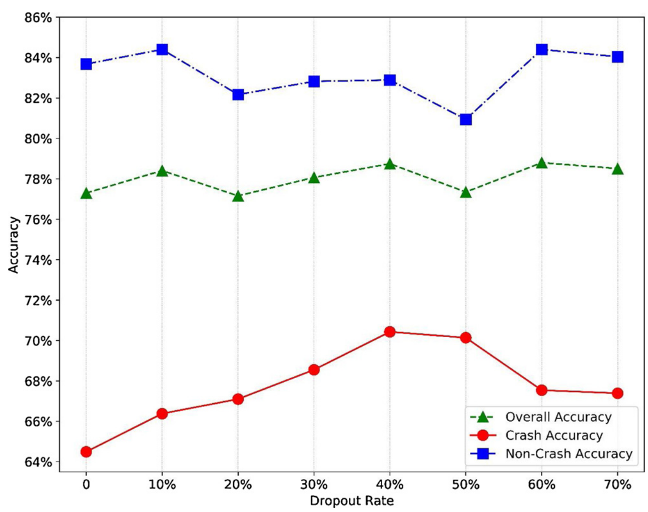

Fig. 12. Results of model with different dropout rates.

LSTMDTR model obtains satisfactory performance on crash detection.

(2) Six LSTMDTR models are constructed for crash detection and obtain desirable performance. Especially, the model with traffic data of speed, flow, and occupancy from six nearest stations can achieve the highest crash accuracy of 70.43 %.
(3) LSTMDTR models constructed on one freeway can be transferred to predict crashes on other similar freeways as long as proper features and networks are provided. Especially, the transferability model with traffic data of speed, flow, and occupancy from six nearest stations can detect crashes with the highest crash accuracy of 65.12 %.
(4) Compared with machine learning methods (e.g., KNN, LR, NB, ANN, SVM, and RF), the LSTMDTR model shows better prediction performance on crash detection and model transferability. This indicates that the importance of applying deep learning methods on crash detection has been overlooked in previous studies.
(5) LSTMDTR model performs better than LSTM models with traffic data of one or two temporal resolutions. This indicates that the importance of considering traffic data of three different temporal resolutions has been overlooked in the previous studies.
(6) The number of neurons can affect prediction performance and computation time on the LSTMDTR model, with crash accuracy ranging from 63.48 % to 70.43 %. A proper number of neurons should be determined in real applications considering acceptable detection performance and computational time.
(7) The dropout technique can reduce overfitting and improve the generalization ability of the LSTMDTR model, increasing crash accuracy from 64.49 % to 70.43 %. It is important to optimize the dropout rate in real applications to obtain optimal performance of detection models.

The limitation of this study is that cases with very poor data quality (e.g., no data recorded in more than one stations) are deleted in data preprocessing. However, this kind of missing data accounts for a large proportion of all cases. Future work needs to propose proper methods to supplement these missing data and improve prediction performance.

CRediT authorship contribution statement

Feifeng Jiang: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Kwok Kit Richard Yuen:** Supervision, Project administration, Funding acquisition. **Eric Wai Ming Lee:** Supervision, Project administration.

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.

Acknowledgment

This work was fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11301015 and Project No. T32-101/15-R). The authors wish to appreciate the anonymous reviewers for their comments to improve this paper.

References

Abdel-Aty, M., Pande, A., 2005. Identifying crash propensity using specific traffic speed conditions. J. Saf. Res. 36, 97–108. https://doi.org/10.1016/j.jsr.2004.11.002.
Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, M.F., Hsia, L., 2004. Predicting freeway crashes from loop detector data by matched case-control logistic regression. Transp. Res. Rec. 1897, 88–95. https://doi.org/10.3141/1897-12.
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. https://doi.org/10.1016/j.trc.2012.04.001.
Ahmed, M.M., Abdel-Aty, M.A., 2012. The viability of using automatic vehicle identification data for real-time crash prediction. IEEE Trans. Intell. Transp. Syst. 13, 459–468. https://doi.org/10.1109/TITS.2011.2171052.
Cheng, J.C.P., Ma, L.J., 2015a. A non-linear case-based reasoning approach for retrieval of similar cases and selection of target credits in LEED projects. Build. Environ. 93, 349–361. https://doi.org/10.1016/j.buildenv.2015.07.019.
Cheng, J.C.P., Ma, L.J., 2015b. A data-driven study of important climate factors on the achievement of LEED-EB credits. Build. Environ. 90, 232–244. https://doi.org/10.1016/j.buildenv.2014.11.029.
Federal Highway Administration, 2004. Traffic Incident Management.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
Hossain, M., Abdel-Aty, M., Quddus, M.A., Muromachi, Y., Sadeek, S.N., 2019. Real-time crash prediction models: state-of-the-art, design pathways and ubiquitous requirements. Accid. Anal. Prev. 124, 66–84. https://doi.org/10.1016/j.aap.2018.12.022.
Hossain, M., Muromachi, Y., 2012. A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways. Accid. Anal. Prev. 45, 373–381. https://doi.org/10.1016/j.aap.2011.08.004.
Lee, C., Hellinga, B., Saccomanno, F., 2003. Real-time crash prediction model for application to crash prevention in freeway traffic. Transp. Res. Rec. 1840, 67–77. https://doi.org/10.3141/1840-08.
Lei, J., Liu, C., Jiang, D., 2019. Fault diagnosis of wind turbine based on Long Short-term memory networks. Renew. Energy 133, 422–432. https://doi.org/10.1016/j.renene.2018.10.031.
Liu, M., Chen, Y., 2017. Predicting real-time crash risk for urban expressways in China. Math. Probl. Eng. 2017, 1–10. https://doi.org/10.1155/2017/6263726.
Liu, Q., Lu, J., Chen, S., Zhao, K., 2014. Multiple Naïve Bayes classifiers ensemble for traffic incident detection. Math. Probl. Eng. 2014, 1–16. https://doi.org/10.1155/2014/383671.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16, 865–873. https://doi.org/10.1109/TITS.2014.2345663.
Ma, J., Cheng, J.C.P., 2016. Estimation of the building energy use intensity in the urban scale by integrating GIS and big data technology. Appl. Energy 183, 182–192.

https://doi.org/10.1016/j.apenergy.2016.08.079.
Ma, J., Ding, Y., Cheng, J.C.P., Jiang, F., Wan, Z., 2019a. A temporal-spatial interpolation and extrapolation method based on geographic Long Short-Term Memory neural network for PM2.5. J. Clean. Prod. 237, 117729. https://doi.org/10.1016/j.jclepro.2019.117729.
Ma, J., Ding, Y., Cheng, J.C.P., Tan, Y., Gan, V.J.L., Zhang, J., 2019b. Analyzing the leading causes of traffic fatalities using XGBoost and grid-based analysis: a city management perspective. IEEE Access 7, 148059–148072. https://doi.org/10.1109/ACCESS.2019.2946401.
Ma, J., Ding, Y., Gan, V.J.L., Lin, C., Wan, Z., 2019c. Spatiotemporal prediction of PM2.5 concentrations at different time granularities using IDW-BLSTM. IEEE Access 7, 107897–107907. https://doi.org/10.1109/ACCESS.2019.2932445.
Ma, J., Li, Z., Cheng, J.C.P., Ding, Y., Lin, C., Xu, Z., 2019d. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. Sci. Total Environ. 135771. https://doi.org/10.1016/j.scitotenv.2019.135771.
Motamed, M., 2016. Developing a Real-Time Freeway Incident Detection Model Using Machine Learning Techniques (Thesis). https://doi.org/10.15781/T28G8FJ1F.
National Highway Traffic Safety Administration, 2016. Traffic Safety Facts 2016.
Oh, C., Oh, J.-S., Ritchie, S.G., 2005. Real-time hazardous traffic condition warning system: framework and evaluation. IEEE Trans. Intell. Transp. Syst. 6, 265–272. https://doi.org/10.1109/TITS.2005.853693.
Pande, A., Abdel-Aty, M., 2006. Assessment of freeway traffic parameters leading to lane-change related collisions. Accid. Anal. Prev. 38, 936–948. https://doi.org/10.1016/j.aap.2006.03.004.
Pande, A., Nuworsoo, C., Shew, C., 2012. Proactive Assessment of Accident Risk to Improve Safety on a System of Freeways (No. MTI Report 11–15). Transportation Policy Studies.
Parsa, A.B., Taghipour, H., Derrible, S., Mohammadian, A.K., 2019. Real-time accident detection: coping with imbalanced data. Accid. Anal. Prev. 129, 202–210. https://doi.org/10.1016/j.aap.2019.05.014.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 1929–1958.
Sun, Jian, Sun, Jie, Chen, P., 2014. Use of support vector machine models for real-time prediction of crash risk on urban expressways. Transp. Res. Rec. 2432, 91–98. https://doi.org/10.3141/2432-11.
Sun, Jie, Sun, Jian, 2015. A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data. Transp. Res. Part C Emerg. Technol. 54, 176–186. https://doi.org/10.1016/j.trc.2015.03.006.
Wang, Shouxiang, Wang, X., Wang, Shaomin, Wang, D., 2019. Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. Int. J. Electr. Power Energy Syst. 109, 470–479. https://doi.org/10.1016/j.ijepes.2019.02.022.
Wunnava, S., Qin, X., Kakar, T., Rundensteiner, E.A., Kong, X., 2018. Bidirectional LSTM-CRF for Adverse Drug Event Tagging in Electronic Health Records. International Workshop on Medication and Adverse Drug Event Detection, pp. 48–56.
Xiao, J., 2019. SVM and KNN ensemble learning for traffic incident detection. Phys. A Stat. Mech. Appl. 517, 29–35. https://doi.org/10.1016/j.physa.2018.10.060.
Xu, C., Wang, W., Liu, P., 2013. A genetic programming model for real-time crash prediction on freeways. IEEE Trans. Intell. Transp. Syst. 14, 574–586. https://doi.org/10.1109/TITS.2012.2226240.
Xu, C., Wang, W., Liu, P., Guo, R., Li, Z., 2014. Using the Bayesian updating approach to improve the spatial and temporal transferability of real-time crash risk prediction models. Transp. Res. Part C Emerg. Technol. 38, 167–176. https://doi.org/10.1016/j.trc.2013.11.020.
Yao, B., Hu, P., Zhang, M., Jin, M., 2014. A support vector machine with the tabu search algorithm for freeway incident detection. Int. J. Appl. Math. Comput. Sci. 24, 397–404. https://doi.org/10.2478/amcs-2014-0030.
Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. Accid. Anal. Prev. 51, 252–259. https://doi.org/10.1016/j.aap.2012.11.027.
Yu, R., Wang, X., Yang, K., Abdel-Aty, M., 2016. Crash risk analysis for Shanghai urban expressways: a Bayesian semi-parametric modeling approach. Accid. Anal. Prev. Traffic Saf. China: Challenges Countermeasures 95, 495–502. https://doi.org/10.1016/j.aap.2015.11.029.
Yuan, F., Cheu, R.L., 2003. Incident detection using support vector machines. Transp. Res. Part C: Emerg. Technol. Traffic Detect. Estimation 11, 309–328. https://doi.org/10.1016/S0968-090X(03)00020-2.
Yuan, J., Abdel-Aty, M., 2018. Approach-level real-time crash risk analysis for signalized intersections. Accid. Anal. Prev. 119, 274–289. https://doi.org/10.1016/j.aap.2018.07.031.