2020

A Deep Learning Approach for Real-time Crash Risk Prediction at Urban Arterials

Pei Li
University of Central Florida

A DEEP LEARNING APPROACH FOR REAL-TIME CRASH RISK PREDICTION AT URBAN ARTERIALS

by

PEI LI

B.Sc. Tongji University, 2015

M.Sc. Tongji University, 2018

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Civil, Environmental and Construction Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term

2020

Major Professor: Mohamed Abdel-Aty

ABSTRACT

Real-time crash risk prediction aims to predict the crash probabilities within a short time period, it is expected to play a crucial role in the advanced traffic management system. However, most of the existing studies only focused on freeways rather than urban arterials because of the complicated traffic environment of the arterials. This thesis proposes a long short-term memory convolutional neural network (LSTM-CNN) to predict the real-time crash risk at arterials. The advantage of this model is it can benefit from both LSTM and CNN. Specifically, LSTM captures the long-term dependency of the data while CNN extracts the time-invariant features. Four urban arterials in Orlando, FL are selected to conduct a case study. Different types of data are utilized to predict the crash risk, including traffic data, signal timing data, and weather data. Various data preparation techniques are applied also. In addition, the synthetic minority over-sampling technique (SMOTE) is used for oversampling the crash cases to address the data imbalance issue. The LSTM-CNN is fine-tuned on the training data and validated on the test data via different metrics. In the end, five other benchmarks models are also developed for model comparison, including Bayesian Logistics Regression, XGBoost, LSTM, CNN, and Sequential LSTM-CNN. Experimental results suggest that the proposed LSTM-CNN outperforms others in terms of Area Under the Curve (AUC) value, sensitivity, and false alarm rate. The findings of this thesis indicate the promising performance of using LSTM-CNN to predict real-time crash risk at arterials.

Keywords: Real-time crash risk, urban arterials, recurrent neural network, deep learning

ACKNOWLEDGMENT

I would like to convey my heartiest gratitude to my honorable supervisor Dr. Mohamed Abdel-Aty for his excellent supervision and constant support in this thesis. I would like to acknowledge Dr. Yuan for his guidance and support. I would also like to acknowledge the support and encouragement from my family and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AUC: Area under Curve

CNN: Convolutional neural network

FPR: False positive rate

LSTM: Long-short term memory neural network

ROC: Receiver operating characteristics

SMOTE: Synthetic minority over-sampling technique

TPR: True positive rate

CHAPTER 1: INTRODUCTION

1.1 Introduction

According to the data from National Highway Traffic Safety Administration (NHTSA), in 2018, traffic crashes have caused 33,654 fatalities in the USA, while 41.6% of them happened at urban arterials. To enhance traffic safety, many studies have been conducted to help the development of the Active Traffic Management Systems. One of its crucial functions is to predict the crash risk in real-time, which can help the traffic managers make decisions proactively. Different from the traditional crash frequency prediction based on aggregated data, real-time crash risk prediction aims to predict crash probabilities during a short time period (e.g. 15 minutes or 20 minutes). However, one difficulty for real-time crash risk prediction is it requires lots of real-time data as the input. Recently, with the implementation of various intelligent transportation technologies, extensive real-time traffic data became available through fixed sensors, vehicle detectors, etc. All of these makes it is possible to predict crash risk in real-time.

There are lots of studies on the real-time crash risk prediction at freeways (M. A. Abdel-Aty, Hassan, Ahmed, & Al-Ghamdi, 2012; Ahmed, Abdel-Aty, & Yu, 2012; Oh, Oh, Ritchie Stephen, & Chang, 2005; Xu, Tarko, Wang, & Liu, 2013; Yu & Abdel-Aty, 2014). However, the studies on urban arterials are still limited (Theofilatos, 2017; Yuan et al., 2018). Moreover, deep learning methods are now widely applied to transportation fields. There are lots of studies on traffic speed prediction, traffic volume prediction, etc. Nevertheless, one few studies applied deep learning methods on traffic safety analysis. Considering this research gaps, this thesis wants to investigate the real-time crash risk prediction at the urban arterials based on deep learning methods.

1.2 Thesis Contribution

This thesis has made several contributions to real-time crash risk prediction at urban arterials. Firstly, this thesis is one of the pioneer studies which explores the application of LSTM-CNN on real-time crash risk prediction. Secondly, the possibilities of using various data sources for real-time crash prediction are explored, such as Bluetooth data, signal timing data and weather data. One year's data are analyzed extensively, besides, different data preparation techniques are utilized. Thirdly, the data imbalance problem is addressed based on different data sampling methods. In the end, the performance of LSTM-CNN is compared with other benchmark approaches on the same dataset. Results suggest that the LSTM-CNN outperforms the others with various evaluation metrics, i.e., AUC, sensitivity and false alarm rate.

1.3 Thesis Objectives

The objectives can be summarized as below:

- A deep learning model will be built for predicting real-time crash risk.
- Different data preparation methods will be applied for processing large scale data.
- The possibilities of fusing traffic, signal, and weather data will be explored.
- The performance of proposed model will be compared with different benchmark methods.

1.4 Thesis Structure

The rest of the thesis is organized as followings: Chapter 2 reviews the related studies about real-time crash risk prediction. Chapter 3 introduces the main methodologies used in this study. Chapter 4 presents the preparation steps for traffic, signal, and weather data. Chapter 5 demonstrates the experimental design and results. Also, different methods are compared based on same metrics. In the end, Chapter 6 summaries the main findings from this thesis and future research directions.

CHAPTER 2: LITERATURE REVIEW

2.1 Real-time Crash Risk Prediction

Real-time crash risk prediction aims to predict the crash risk within a short time period. Crash risk prediction is a typical binary classification problem with its output as a categorical event (crash or non-crash). In general, there are two types of studies based on the study area, freeways and urban arterials. The majority of the existing studies are conducted on freeways (M. A. Abdel-Aty et al., 2012; Ahmed et al., 2012; Oh et al., 2005; Yu & Abdel-Aty, 2014). Lots of efforts have been made to investigate the relationships between crash risk and other types of variables. For example, speed standard deviation was found to have significant positive effects on crash occurrence. Higher traffic volume was also positively correlated with the crash risk. Nevertheless, there are only few studies about the real-time crash risk prediction at arterials. Theofilatos (2017) firstly investigated crash likelihood and severity by exploiting real-time traffic and weather collected from urban arterials. He found that both the variation in occupancy and logarithm of the coefficient of variation of flow are positively associated with crash occurrence. However, the traffic parameters were aggregated to 1-hour interval, which might be too large to capture the short-term traffic status prior to crash occurrence. Yuan et al. (2018) examined the relationships between crash occurrence and real-time traffic and signal timing characteristics based on four urban arterials in Central Florida. The authors indicated that the average speed, upstream left-turn volume, downstream green ratio, and rainy indicator were found to have significant effects on crash occurrence.

2.2 Crash Sampling Methodologies

To predict crash risk accurately, one of the big challenges is the rareness of the crash events. In the real life, non-crash cases are much more common than crash cases, which generates an extremely imbalanced data. The crash prediction model usually has a very bad performance if we directly use the original data without any manipulations. One common approach to address this problem is to data resampling. There are two types of methods are available, matched case-control method and machine learning method. Specifically, matched case-control is a traditional under-sampling method (M. Abdel-Aty, Uddin, Pande, Abdalla, & Hsia, 2004). First, the crash cases are selected. For each crash, its location, time of day, and day of the week are selected as the matching factors. Based on these factors, a subpopulation of the non-crash cases is then selected for each crash. A total of *m* non-crash cases is then selected at random from each subpopulation of non-crash cases. The m+1 observations are then included for modeling. However, since the non-crash cases are selected randomly, some valuable information of unselected non-crash cases may be lost. Therefore, several new resampling methods are developed based on machine learning methods. Synthetic minority over-sampling technique (SMOTE) is a popular method (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) which can over-sample the minority class. Specifically, synthetic examples of the minority class are created. These examples are randomly introduced among the minority class and some of their closest k-neighbors, which is shown in Figure 1. Since the SMOTE is only applied to the training dataset. The test dataset can still reflect the real-world situation. The characteristics of non-crash events are remained during the training procedure. There are several types of SMOTE are available, this thesis selects regular SMOTE since its simplicity and good performance on the real-time crash risk prediction (Li, Abdel-Aty, & Yuan, 2020).

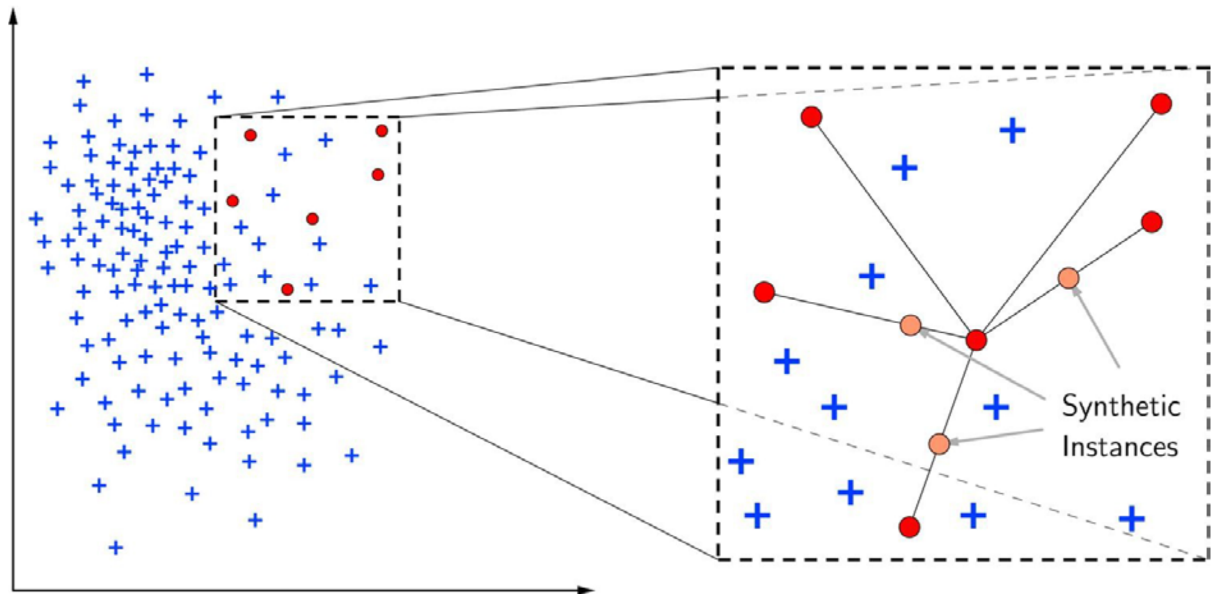


Figure 1 Oversampling with SMOTE (Basso, Basso, Bravo, & Pezoa, 2018)

2.3 Prediction Methodologies

Two kinds of methods are available for real-time crash risk prediction, statistical and machine learning methods. Specifically, statistical methods include conditional logit model, log-linear model, logistic regression, etc. These models are usually built on matched-case control data and have strong assumptions (Shi & Abdel-Aty, 2015; L. Wang, Abdel-Aty, Shi, & Park, 2015; Yu, Wang, Yang, & Abdel-Aty, 2016). Considering these limitations, machine learning methods become popular, such as Support Vector Machine (SVM) (Yu & Abdel-Aty, 2013), Random Forest (L. Lin, Wang, & Sadek, 2015), etc. Recently, with the rapid development of deep learning, it was implemented to solve various transportation problems. Q. Chen, Song, Yamada, and Shibasaki (2016) developed a deep stack denoise autoencoder model to learn from hierarchical features of human mobility and predict traffic accident in aggregated way. Xiaolei Ma et al. (2017) utilized a deep CNN model to predict traffic speed in Beijing. Spatial and temporal traffic dynamic are converted to images describing the time and space relations of traffic flow. Results shown CNN outperformed other methods such as Random Forest and K-Nearest Neighbor. Moreover, RNN was proved to be especially useful for learning time-series data (Y. Tian & Pan, 2015; Zhao, Chen, Wu, Chen, & Liu, 2017; Zheng, Yang, Liu, Dai, & Zhang, 2019). Different from the traditional neural network that only maps the current input vector to output vector (Y. Tian & Pan, 2015), RNN introduces recurrent connections, which allow information to persist. However, one drawback of the RNN is it cannot capture long-term dependency (Bengio, Simard, & Frasconi, 1994). 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 several existing studies that applied LSTM in the transportation field. Zhao et al. (2017) investigated short-term traffic forecast of Beijing based on LSTM, which considers temporal–spatial correlation in the traffic system via a two-dimensional network. The results of LSTM are better than other methods, such as autoregressive integrated moving average model and normal RNN. Similarly, Y. Tian and Pan (2015) utilized LSTM to predict short-term traffic flow. With the ability to memorize long historical data and automatically determine the optimal time lags, LSTM outperformed others such as SVM and single layer feed forward neural network. There are currently few studies used LSTM for real-time crash risk prediction. For example, Yuan, Abdel-Aty, Gong, and Cai (2019) utilized LSTM to predict crash risk in real-time, the authors claimed that their models achieved 60.67% sensitivity, which was much better than the conditional logistic model.

Furthermore, hybrid neural network, which combines the strengths of different neural networks, draws more and more attention recently. Several previous studies combined LSTM and CNN to address temporal and spatial relationships, respectively. For example, Bao, Liu, and Ukkusuri (2019) implemented a spatiotemporal convolutional long short-term memory network to predict the citywide crash frequency based on multiple data sources, such as taxi trip data, road network attributes, and land use features. Similarly, Liu, Zheng, Feng, and Chen (2017) used a LSTM-CNN model to predict short-term traffic flow. Xiaolei Ma et al. (2017) applied a convolutional LSTM network for traffic speed prediction in Beijing. Results from these studies proved the combination of LSTM and CNN achieved better results compared to single LSTM and CNN. However, the exploitation of LSTM-CNN is limited for time-series data, which has no spatial relationship (T. Lin, Guo, & Aberer, 2017). The LSTM and CNN can learn time series data in different ways. LSTM is specialized for learning long-term dependencies and sequential correlations (Zhou, Sun, Liu, & Lau, 2015), while CNN can extract patterns of local trend and the same patterns which appears in different time (T. Lin et al., 2017; C. Tian, Ma, Zhang, & Zhan, 2018). There are only few studies about the application of LSTM-CNN on time series data. Karim, Majumdar, Darabi, and Chen (2018) designed a LSTM-CNN model for time series classification and test its performance on 85 UCR datasets. Results showed this model outperforms other benchmark methods with higher classification accuracy. Furthermore, T. Lin et al. (2017) proposed the TreNet model combined LSTM and CNN. The CNN model is used to extract salient features from local raw data, while LSTM is used to capture long-term dependency. TreNet also achieved better results for time series prediction compared to single CNN and LSTM. To conclude, the good performance of LSTM-CNN on time-series data is achieved by learning local trend and long-term dependency separately, which could complement each other. Real-time crash risk prediction involves lots of time-series data, such as traffic volume, signal timing, traffic speed, etc. The unique architecture of the LSTM-CNN can better capture the long-term and short-term characteristics of these data. Thus, it is promising to investigate the performance of LSTM-CNN on real-time crash risk prediction.

CHAPTER 3: METHODOLOGIES

Three methods are introduced in this part, including LSTM, CNN, and LSTM-CNN. The characteristics of each method will be explained and compared. These models are trained based on the training data after over-sampling and evaluated through test data.

3.1 LSTM

LSTM is one kind of RNN. It is a powerful deep learning method for time series data since its unique design (Hochreiter & Schmidhuber, 1997). RNN suffers from the problem of vanishing gradients, which happens to learning of long data sequences (Aebel, 2018). LSTM can solve it by introducing the memory cell to determine when to forget certain information. A LSTM network is composed of the input layer, the hidden layers, and the output layer. The main characteristics of the LSTM are the memory cells in its hidden layers, which contain memory blocks rather than traditional neuron nodes (Olah, 2015). Each block has several self-connected memory cells and three multiplicative units, input, output and forget gates. These gates provide continuous analogues of write, read and reset operations on the cells. The structure of a LSTM unit at each time step is shown in Figure 2.

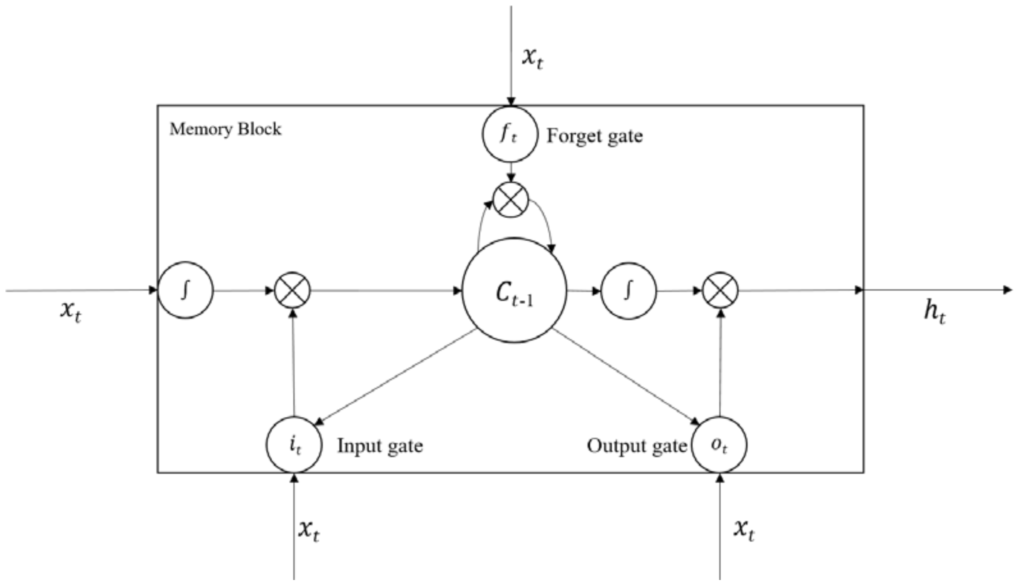


Figure 2 LSTM unit structure (Graves, Mohamed, & Hinton, 2013)

The LSTM generates a mapping from an input sequence vectors $X = (X_1,\ X_2,\ \cdots\ ,\ X_{\mathrm{N}})$ to an output probability vector by calculating the network unit activations using the following equations (Graves et al., 2013), iterated from $t = 1$ to N:

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

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_{fx}X_t} + \boldsymbol{W_{hf}h_{t-1}} + \boldsymbol{W_{cf}c_{t-1}} + \boldsymbol{b_f}) \tag{2}$$

$$\boldsymbol{o_t} = \boldsymbol{\sigma}(\boldsymbol{W_{ox}X_t} + \boldsymbol{W_{oh}h_{t-1}} + \boldsymbol{W_{oc}c_t} + \boldsymbol{b_o}) \tag{3}$$

$$\boldsymbol{c_t} = \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \odot \boldsymbol{tanh}(\boldsymbol{W_{cx}X_t} + \boldsymbol{W_{ch}h_{t-1}} + \boldsymbol{b_c}) \tag{4}$$

$$\boldsymbol{h_t} = \boldsymbol{o_t} \odot \boldsymbol{tanh}(\boldsymbol{c_t}) \tag{5}$$

$$\boldsymbol{y_t} = \boldsymbol{W_{yh}h_{t-1}} + \boldsymbol{b_y} \tag{6}$$

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 ⊙ indicates elementwise product of the vectors. The forget gate f_t controls the extent to which the previous step memory cell is forgotten, the input gate i_t determines how much to update for each unit, and the output gate o_t controls the exposure of the internal memory state. Since the values of all the gating variables vary for each time step, the model could learn how to represent information over multiple time steps.

3.2 CNN

Convolutional neural network (CNN) was originally developed for image classification problems, where the model learns an internal representation of a two-dimensional input, in a process referred to as feature learning (Krizhevsky, Sutskever, & Hinton, 2012). The same process can also be utilized for time series classification tasks and achieve good results (Cui, Chen, & Chen, 2016; Z. Wang, Yan, & Oates, 2017).

The key component for CNN is its convolution layer, where CNN applies a filter to extract features through the input data. Since time series data have one dimension (time), the filter of CNN has also one dimension instead of two dimensions for image (width and height) (Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019). Features generated by a filter usually go through an activation function, such as Rectified Linear Unit (ReLU). The purpose is to introduce non-linearity since most of the real-world data do not always have the linear relationship. After these two procedures, the new features can go through multiple filters and activate functions. The generated features contain valuable information for the prediction or classification problem. There are other useful layers for CNN. For instance, pooling layer is used to reduce the number of parameters, while dropout layer can prevent the over-fitting problem.

Overall, one big advantage for the application of CNN on time series data is its convolutional layer, where the filter is applied to the data across all the time stamps. This allows CNN to learn features that are invariant across the time dimension.

3.3 LSTM-CNN

Both LSTM and CNN have its unique characteristics. Thus, it becomes reasonable to combine them for better results. Previous studies proved the performance of LSTM could be improved by augmenting it with the CNN for time series classification problem. Karim et al. (2018) designed a LSTM-CNN network and applied it on University of California Riverside (UCR) Benchmark datasets. Results indicated that it could achieve state-of-the-art performance compared with other methods. The LSTM and CNN receive the same time-series data as input and their results are concatenated to generate output. The reason is LSTM is capable to learn long-term dependency and CNN can extract time-invariant features. Thus, the combination of them improve the overall accuracy.

The proposed model in this thesis follows the similar logic of the previous studies. It has one LSTM part and one CNN part, the features extracted from them are combined to generate the output. Its structure is elaborated in the following sections.

CHAPTER 4: DATA DESCRIPTION AND PREPARATION

4.1 Data Description

This thesis focuses on the real-time crash risk prediction on urban arterials, more specifically, road segments. A segment is defined as the road facility between two consecutive intersections with the certain direction. Considering the availability of data, eight miles of arterials (38 segments, 21 intersections) are selected from four urban arterials in Orlando, Florida, as shown in Figure 3. Four types of data are collected from September 2017 to September 2018, including crash data provided by Signal Four Analytics (S4A), signal timing, queue length, waiting time and traffic volume provided by the adaptive signal controllers (InSync), vehicle speed data collected through Bluetooth detectors (BlueMAC), and weather data of Orlando International Airport archived by the National Oceanic Atmospheric Administration (NOAA).

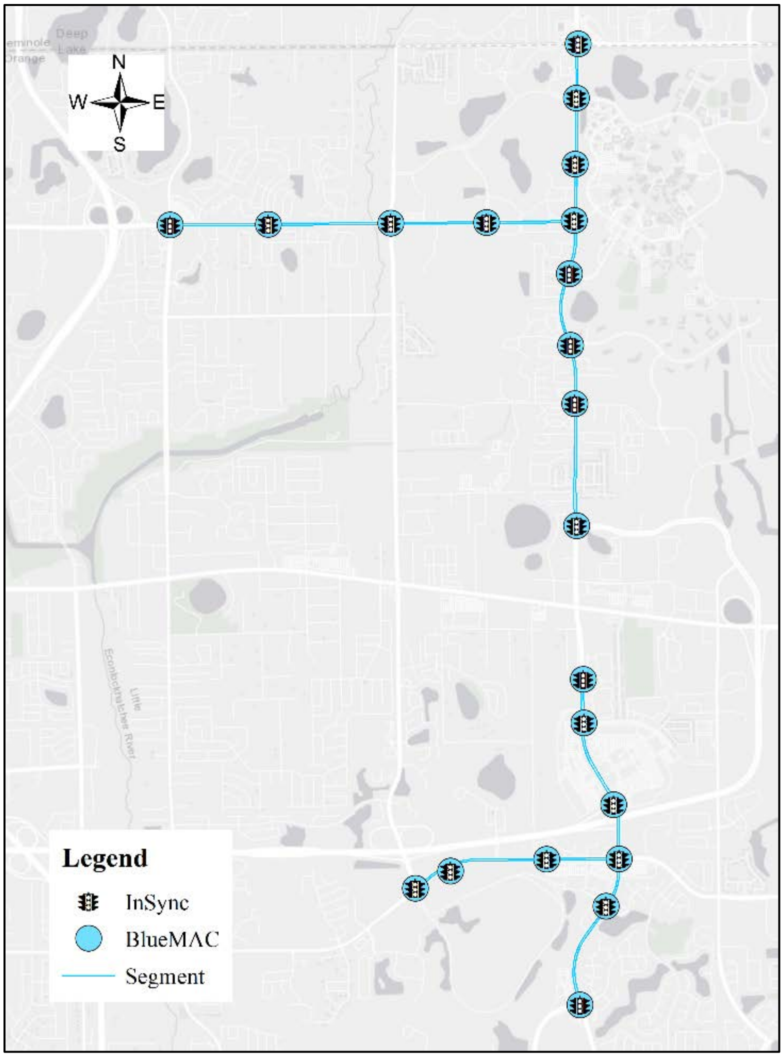


Figure 3 Research area

S4A provides detailed information for every crash event, including crash time, location, severity, and type. Since this thesis only focuses on segment crashes, thus, the crashes within the intersection influence area (within 250 feet of intersection) are excluded from the original dataset. In addition, crashes that resulted from drugs and alcohol are also deleted, since these kinds of crashes are usually not attributed to real-time traffic and signal characteristics which are the focus of this study. After these processes, there are 110 crashes in total. They are allocated to the corresponding segments considering the location of the crash and direction of the segment.

Vehicle speed data are provided by BlueMAC. BlueMAC devices detect the vehicles equipped with Bluetooth devices which are working on discoverable mode, as shown in Figure 4. The individual vehicular speed on a specific segment is calculated as the segment length divided by the travel time of each detected vehicle on this segment. In this thesis, the Bluetooth penetration rate is 3.69%, which is higher than the threshold suggested by the previous studies. Also, the validity of Bluetooth detectors for measuring individual vehicular speed on urban arterials has been proven by previous research (Gong, Abdel-Aty, & Park, 2019; Yuan & Abdel-Aty, 2018).

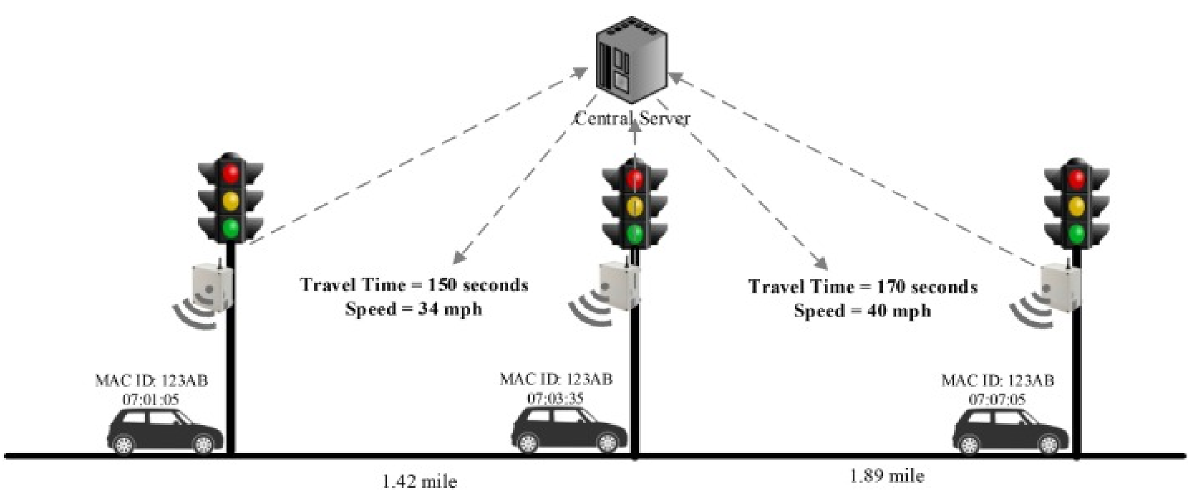


Figure 4 Illustration of Bluetooth data collection (Yuan & Abdel-Aty, 2018)

InSync controller archived the real-time signal timing and lane-specific 15-minute aggregate traffic volume data. The lane-specific 15-minute aggregated traffic volume data are collected by the video detectors, which are installed for the adaptive signal controller to detect the real-time volume, queue length and waiting time. The lane-specific traffic volume for each minute is calculated based on the assumption that the traffic volume within the 15-minute interval is evenly distributed. Furthermore, since a segment has its direction, it has the upstream and the downstream intersections. Traffic features of the two intersections are included in the features of this segment.

Five weather-related variables are collected through NOAA. Since the study area is within 20 miles of the selected airport, the weather data is valid according to the previous research (Chung, Abdel-Aty, & Lee, 2018). In total, each segment has 24 features. Since this thesis aims to predict real-time crash risk in five-minute time-slice and update every minute. All the statistical features are generated in five-minute interval, which are shown in Table 1. Overall, one segment has roughly 525,600 samples (365 days * 24 hours * 60 minutes). It should be noted that this number is overestimated since some segments do not have the whole year's data.

Table 1 Feature description

Type	Feature	Description	Mean (Std)	(Min Max)
Traffic Data	speed_avg	Average speed on the segment in mph	31.45 (11.60)	(0.66 80.00)
	speed_std	Speed standard deviation on the segment	4.35 (5.85)	(0.00 69.38)
	up_lt_volume	Average left turn volume of upstream intersection	4.18 (9.94)	(0.00 309.25)
	up_th_volume	Average through volume of upstream intersection	45.94 (60.65)	(0.00 1363.00)
	down_lt_volume	Average left turn volume of downstream intersection	6.93 (13.38)	(0.00 684.00)
	down_th_volume	Average through volume of downstream intersection	43.74 (56.32)	(0.00 1208.00)
	up_lt_queue	Average maximum left turn queue length of upstream intersection	2.76 (4.87)	(0.00 99.00)
	up_th_queue	Average maximum through queue length of upstream intersection	1.30 (3.97)	(0.00 40.00)
	down_lt_queue	Average maximum left turn queue length of downstream intersection	2.65 (4.92)	(0.00 99.00)
	down_th_queue	Average maximum through queue length of downstream intersection	1.65 (3.68)	(0.00 40.09)
	up_lt_wait	Average maximum left turn wait time of upstream intersection	9.97 (14.29)	(0.00 55.50)
	up_th_wait	Average maximum through wait time of upstream intersection	8.71 (20.71)	(0.00 38.46)
	down_lt_wait	Average maximum left turn wait time of downstream intersection	9.46 (13.91)	(0.00 63.96)
	down_th_wait	Average maximum through wait time of downstream intersection	15.50 (17.79)	(0.00 37.72)
Signal Timing Data	up_lt_green_time_ratio	Ratio of left turn green time of upstream intersection	5.39 (6.86)	(0.00 30.00)
	up_th_green_time_ratio	Ratio of through green time of upstream intersection	33.49 (30.96)	(0.00 100.00)
	down_lt_green_time_ratio	Ratio of left turn green time of downstream intersection	2.77 (5.37)	(0.00 40.00)
	down_lt_green_time_ratio	Ratio of through green time of downstream intersection	32.52 (30.26)	(0.00 100.00)
Weather Data	visibility	Horizontal distance an object can be seen and identified given in miles	9.64 (1.53)	(0.00 10.00)
	weathertype	Normal weather: 0. Abnormal weather data: 1	0.06 (0.24)	(0.00 1.00)
	humidity	Relative humidity given in percentage	74.13 (20.54)	(9.00 100.00)
	precipitation	Amount of precipitation in inches to hundredths	0.00 (0.04)	(0.00 1.99)
	temperature	Dry-bulb temperature in whole degrees Fahrenheit	70.72 (11.45)	(0.00 94.00)
	windspeed	Speed of the wind at the time of observation given in mph	8.47 (4.87)	(0.00 56.00)

4.2 Data Preparation

4.2.1 Feature Selection

Feature importance and correlation were addressed based on extra-tree (Geurts, Ernst, & Wehenkel, 2006) and Pearson correlation coefficient (Benesty, Chen, Huang, & Cohen, 2009). After normalizing all the features according to the maximum and minimum values, the results of feature importance and feature correlation are shown in Figure 5, respectively. Extra-tree classifier implements a meta estimator that 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). Pearson correlation coefficient measures the correlation between every pair of features from -1 to 1. It reflects the strength of the relationship between two variables. In general, a correlation is treated as strong if its absolute value is larger than 0.5 (Cohen, 1992).

Several features have high correlations with others, such as ‘visibility’ and ‘humidity’. Moreover, some features have extremely lower importance than others, such as ‘weather type’ and ‘visibility. A feature selection rule is made according to these two findings. A feature will be eliminated if it is highly correlated with another feature but are less important than the other one. In the end, 11 features were included in the final dataset, including speed_avg, speed_std, up_th_volume, down_th_volume, down_th_queue, down_lt_queue, down_lt_wait, up_th_green_time_ratio, humidity, temperature, windspeed.

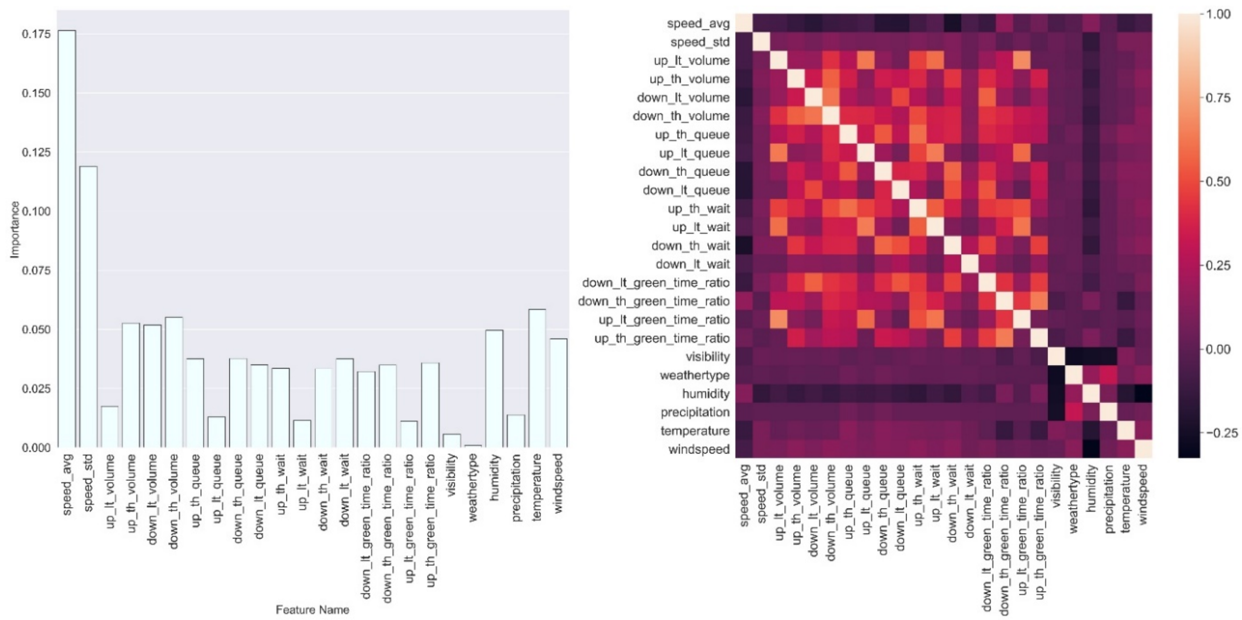


Figure 5 Feature importance and correlation

4.2.2 Crash Labeling

In this thesis, data of three time-slice are stacked to predict the crash risk during next 5-10 minutes based on LSTM-CNN. To create ‘crash’ label for samples in every minute, they are labeled according to actual crash events (Figure 6). For instance, if a crash happens at 00:10, the traffic safety statuses from 00:00 to 00:05 are then labelled as ‘1’, indicating that a crash will occur in the next 5-10 minutes. While the other data outside of the 00:00 to 00:05 are labeled as ‘0’. In addition, data within 120 minutes after a crash are removed since the crash usually causes turbulence.

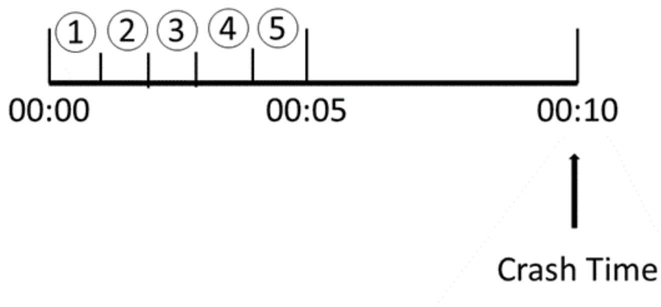


Figure 6 Crash labeling illustration

After data preparation and crash labeling, there are 7098269 non-crash events and 432 crash events. The crash to non-crash ratio is around 1:16,431, indicating that the data are extremely imbalanced. There are many methods to solve this problem. For example, coping more minority classes or adjusting the cost function to make misclassification of minority classes more important than misclassification of majority instances. In this thesis, a resampling technique is employed to tackle this problem. More specifically, over-sampling, which creates more samples for the minority classes. Synthetic Minority Over-sampling Technique (SMOTE) is a powerful method among existing studies (Chawla et al., 2002). SMOTE uses a nearest neighbors' algorithm to generate new and synthetic data. It synthesizes new minority instances between real minority instances. In this thesis, we use SMOTE to generate synthetic samples of crash events to create an equal number between crash and non-crash events. Besides, SMOTE is only applied to the training data, while the test data are still the real-world data.

CHAPTER 5: EXPERIMENTS AND RESULTS

5.1 Network Architecture

The network architecture of our LSTM-CNN is shown in Figure 7. It has two LSTM layers and two CNN layers. Moreover, dropout layers are added to prevent over-fitting, while average pooling layer is included for reducing the number of parameters. The LSTM and CNN are then combined by a concatenate layer.

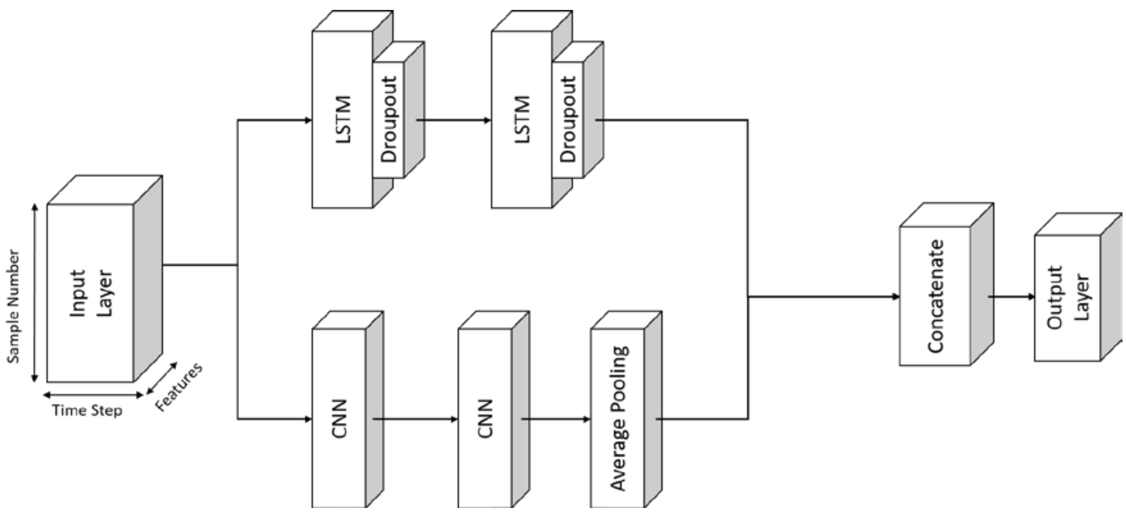


Figure 7 Network architecture

The input of the network is prepared with the shape as (S, T, F), where S is the sample size, T is the time-slice and F is the number of features. In this thesis, we prepared the dataset as (S, 3, 11) since we stack the data of three time-slice and have 11 features. The CNN and LSTM modules receive data simultaneously and learned it separately. However, there are also several papers that indicated that the LSTM-CNN can be prepared in a sequential way (X. Ma, Zhang, Du, Ding, & Sun, 2019; J. Zhang, Ma, Ding, Wang, & Liu, 2018). Which means that the CNN module receives the input first and its results are then passed to the LSTM module. The performance of parallel and sequential LSTM-CNN would be compared in the following sections.

5.2 Experiment Design and Results

The experiment procedure is shown in Figure 8 . First, the data are divided into training (75%) and test (25%). Second, the proposed model is trained based on training data after over-sampling. In the end, the trained model is evaluated according to the test data.

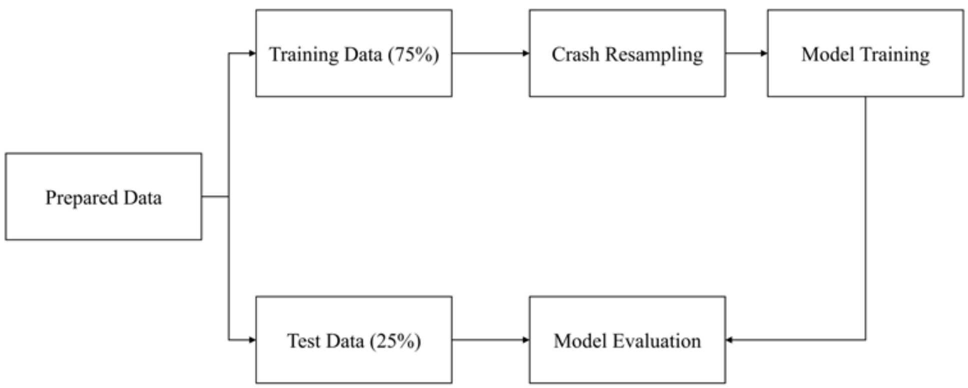


Figure 8 Experiment procedure

Regarding model evaluation, AUC (Hanley & McNeil, 1982) is adopted in this thesis, which is the area under the receiver operating characteristics (ROC) curve. Specifically, A ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: true positive rate and false positive rate, which can be found from equations (7) and (8), respectively. AUC measures the entire two-dimensional area underneath the entire ROC curve from (0, 0) to (1, 1). In addition, sensitivity and false alarm rate are also included as other two metrics. Sensitivity is usually referred as True Positive

Rate (TPR), while false alarm rate is also called as False Positive Rate (FPR). The formulations of TPR and FPR are shown as following:

$$\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \tag{7}$$

$$\boldsymbol{FPR} = \frac{\boldsymbol{FP}}{\boldsymbol{FP} + \boldsymbol{TN}} \tag{8}$$

The values of TP (True Positive), FP (False Positive), FN (False Negative), TN (True Negative) are calculated based on the classification confusion matrix (Table 2).

Table 2 Classification confusion matrix

	True Crash	True Non-crash
Predicted Crash	TP	FP
Predicted Non-Crash	FN	TN

The LSTM-CNN model is implemented based on Keras (Chollet, 2015) using NVIDIA GTX 1050 4G GPU. During the training process, one crucial step is to select the proper hyperparameters and optimization functions to achieve the best result. There are several optimization functions are available, such as Stochastic Gradient Descent (SGD), RMSProp, and Adaptive Moment Estimation (Adam). Besides, five other hyperparameters are tuned, including the filter size of CNN module, unit number of LSTM module, learning rate, epoch number, and batch size, which are summarized in Table 3.

Table 3 Hyperparameters tuning

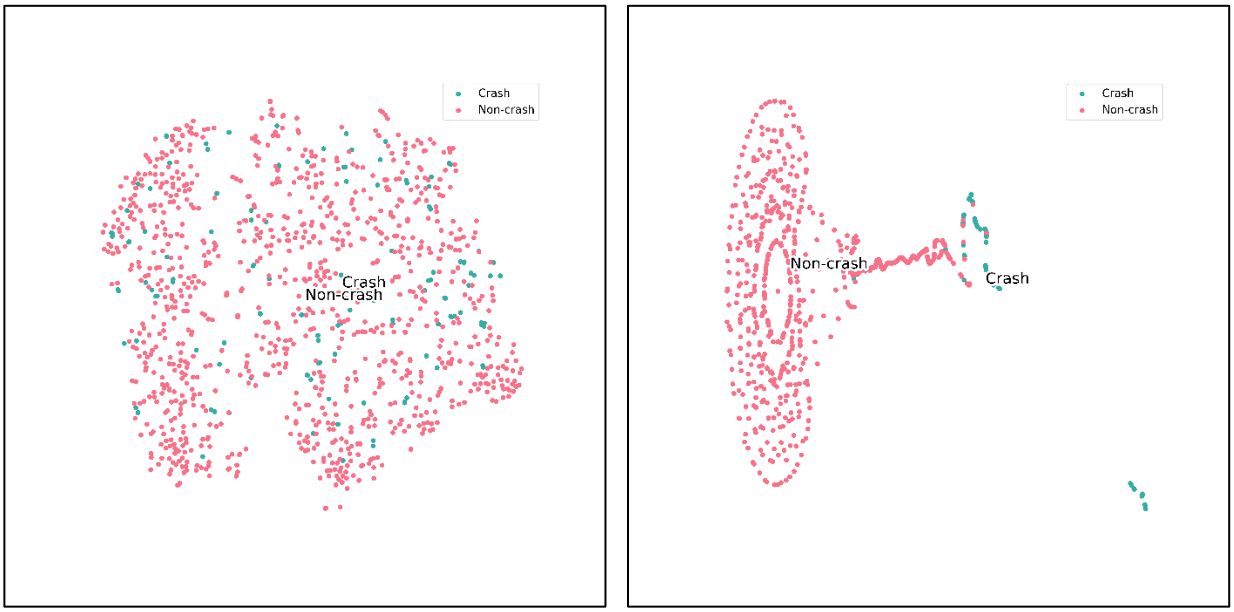
Hyperparameter	Range	Value
Learning Rate	0.0001, 0.001, 0.01	0.01
Epoch Number	50, 100, 150, 200, 500	100
Batch Size	1000, 5000, 10,000, 20,000	10,000
CNN Filter Size	128, 64, 32, 16	64
LSTM Unit Number	128, 64, 32, 16	16
Optimization Function	Adam, RMSprop, SGD	Adam

After the model is fine-tuned. Its results are shown in Table 4. To estimate the sensitivity and false alarm rate, the threshold of ROC curve is chosen as the point where sensitivity (true positive rate) equals to specificity (true negative rate). The proposed model achieves state-of-the-art results with high AUC value, high sensitivity, and low false alarm rate.

Table 4 Experiment results

Metric Name	AUC	False Alarm Rate	Sensitivity
Value	0.932	0.132	0.868

Furthermore, to illustrate the model's capacity of distinguishing crash and non-crash events, t-Distributed Stochastic Neighbor Embedding (t-SNE) is introduced to visualize features for crash and non-crash events. t-SNE is a powerful tool for high-dimensional data visualization (Maaten & Hinton, 2008). It maps multi-dimensional data to two or more dimensions suitable for human observation. Specifically, t-SNE of the raw features is shown in Figure 9 (a), and t-SNE of the extracted features from the last layer of LSTM-CNN is shown in Figure 9 (b). Obviously, these two events are difficult to distinguish in the raw features, their patterns are critically tangled together (Figure 9 (a)). However, the features extracted from LSTM-CNN successfully divide them. The two events become almost separable. But it is also worth to notice that some non-crash events still mix with crash events. The performance of the model can be improved in the future.



(a) (b)

Figure 9 t-SNEs of raw features and extracted features

5.3 Model Comparison

Five benchmark models are selected to compare the performance with the proposed LSTM-CNN, including Bayesian logistic regression, XGBoost, LSTM, CNN, and Sequential LSTM-CNN.

Bayesian logistic regression model was widely applied to predict crash probability (Ahmed et al., 2012; Yu & Abdel-Aty, 2013). Bayesian logistic regression is built based on the classical logistic equation and can deal with both continuous and categorical variables. The traditional logistic regression usually treats the parameters of the models as fixed, and the data are only used to get a best estimate of the unknown values of the parameters. Differently, the

Bayesian logistics regression treats the parameters 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). Assuming the probability for crash (y = 1) and non-crash (y = 0) is p and 1 – p, respectively. The model can be written as below:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta X \tag{9}$$

Where β_0 is the intercept, $\boldsymbol{X}$ is the vector of the independent variables used in this thesis, such as volume, speed, temperature, etc. $\boldsymbol{\beta}$ is the vector of coefficients for the independent variables.

XGBoost is the short name for eXtreme Gradient Boosting, which is a scalable machine learning system for tree boosting (T. Chen & Guestrin, 2016). XGBoost is an optimized gradient tree boosting system, with some algorithmic innovations (e.g., approximate greedy search, parallel learning) and hyperparameters to improve learning and control over-fitting. This method is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges, as also as many transportation fields.

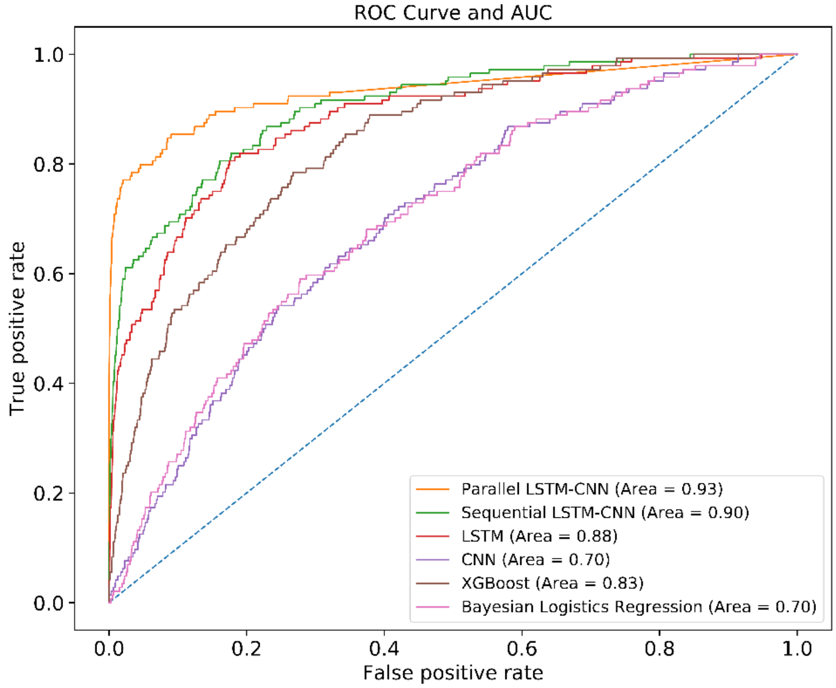
These five models are trained and tested on the same dataset as the proposed model. However, the input of some models is different with the LSTM-CNN, such as the XGBoost and Bayesian Logistic Regression. Therefore, the input data is reshaped to (S, 3 × 11). In addition, feature selection is applied for Bayesian logistics regression considering the correlations between different features. All these models are fine-tuned based on the same training data as the LSTM-CNN and validated on the test data. Figure 10 (a) shows the ROC curves and AUC values of all the six models on the test data. the proposed LSTM-CNN has the highest AUC values of 0.93, which is better than the Sequential LSTM-CNN. Besides, Figure 10 (b) indicates the proposed

LSTM-CNN has the highest sensitivity and lowest false alarm rate, which proves its good performance from another aspect. From the model comparison, several conclusions can be summarized as following:

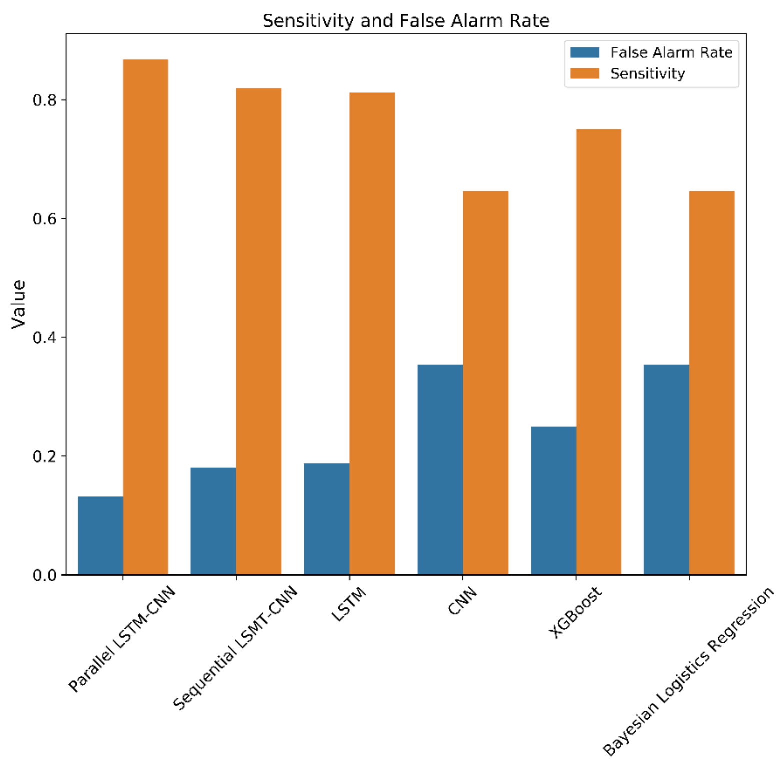
(a) In general, the proposed LSTM-CNN outperforms the other methods in terms of AUC, sensitivity, and false alarm rate. The result confirms the feasibility and superiority of the proposed method, which can accurately predict the real-time crash risk on arterials. In addition, low false alarm rate is one of the most unique advantages of the proposed model.

(b) The performance of the LSTM can be enhanced by augmenting it with the CNN component. It is suggested to prepare the LSTM-CNN in a parallel way rather than a sequential way. Besides, CNN cannot accurately predict the real-time crash alone since it fails to capture the long-term dependency.

(c) Deep learning and machine learning models tend to have better results than statistical models. XGBoost reaches a relatively decent accuracy. Its performance on the real-time crash risk prediction can be exploited in the future.



(a) ROC Curve and AUC



(b) Sensitivity and False Alarm Rate

Figure 10 Model comparison results

CHAPTER 6: CONCLUSIONS

This study applied LSTM-CNN to predict real-time crash risk on urban arterials. First, different data were collected for one year, such as traffic flow, signal timing and weather conditions. Second, raw data were cleaned and pre-possessed according to correlation and importance. Third, SMOTE was applied as an over-sampling technique to solve the imbalanced data problem. In the end, a LSTM-CNN model was built to predict real-time crash risk. Several benchmark models are implemented for comparisons with various evaluation metrics.

The results suggest that the proposed LSTM-CNN outperforms the others from several aspects. First, it achieves the highest sensitivity as 88% and lowest false alarm rate as 12%. Second, it has the highest AUC value as 0.93, which is much higher than other methods, including LSTM, CNN, XGBoost, Bayesian Logistics regression, etc. In the end, the t-SNE plot indicates the features extracted by the LSTM-CNN successfully distinguish crash and non-crash events, which illustrate the reason for its good performance.

Overall, this thesis succeeds in verifying the possibilities to predict real-time crash risk based on deep neural network. Besides, the performance of LSTM can be enhanced by introducing CNN to extract features differently. SMOTE is proved as a useful over-sample method for large imbalanced crash dataset. The results of this thesis can be used for the implementation of an advanced traffic management system, which has the potential to reduce crashes. However, there are still several limitations in the current study. First, the performance of LSTM-CNN can be improved by adding more layers or trying more combinations of different hyperparameters. In addition, geographical features can also be considered as possible inputs for the CNN part, which may help us improve the model's performance. Second, the impact of different time-slice can be tested, the way to prepare the data may influence the model's performance. Third, SMOTE is only one method for over-sampling, there are various methods for the same purpose, such as adaptive boosting and gradient tree boosting. The results of different over-sampling methods are worth to investigate. In the end, our data set lacks driver characteristics. Augmenting this dataset can significantly improve the performance of our model. However, it is hard to get such data. Connected vehicle technology may help in providing more comprehensive driving data in the future (Rahman, Abdel-Aty, Lee, & Rahman, 2019; Yue, Abdel-Aty, Wu, & Wang, 2018; S. Zhang, Abdel-Aty, Yuan, & Li, 2020).

REFERENCES

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. *Transportation Research Record, 1897*(1), 88-95. doi: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. *Transportation Research Part C: Emerging Technologies, 24*, 288-298. doi:10.1016/j.trc.2012.04.001

Aebel, N. (2018). How LSTM networks solve the problem of vanishing gradients. Retrieved from https://medium.com/datadriveninvestor/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

Ahmed, M. M., Abdel-Aty, M., & Yu, R. (2012). Assessment of Interaction of Crash Occurrence, Mountainous Freeway Geometry, Real-Time Weather, and Traffic Data. *Transportation Research Record, 2280*(1), 51-59. doi:10.3141/2280-06

Bao, J., Liu, P., & Ukkusuri, S. V. (2019). A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data. *Accident Analysis & Prevention, 122*, 239-254. doi:https://doi.org/10.1016/j.aap.2018.10.015

Basso, F., Basso, L. J., Bravo, F., & Pezoa, R. (2018). Real-time crash prediction in an urban expressway using disaggregated data. *Transportation Research Part C: Emerging Technologies, 86*, 202-219. doi:https://doi.org/10.1016/j.trc.2017.11.014

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1-4): Springer.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*(2), 157-166. doi:10.1109/72.279181

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research, 16*, 321-357.

Chen, Q., Song, X., Yamada, H., & Shibasaki, R. (2016). *Learning deep representation from big and heterogeneous data for traffic accident inference*. Paper presented at the Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system.* Paper presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.

Chollet, F. (2015). Keras.

Chung, W., Abdel-Aty, M., & Lee, J. (2018). Spatial analysis of the effective coverage of land-based weather stations for traffic crashes. *Applied Geography, 90*, 17-27. doi:https://doi.org/10.1016/j.apgeog.2017.11.010

Cohen, J. (1992). A power primer. *Psychological bulletin, 112*(1), 155.

Cui, Z., Chen, W., & Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery, 33*(4), 917-963.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning, 63*(1), 3-42.

Gong, Y., Abdel-Aty, M., & Park, J. (2019). Evaluation and augmentation of traffic data including Bluetooth detection system on arterials. *Journal of Intelligent Transportation Systems*, 1-13. doi:10.1080/15472450.2019.1632707

Graves, A., Mohamed, A., & Hinton, G. (2013, 26-31 May 2013). *Speech recognition with deep recurrent neural networks.* Paper presented at the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29-36.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2018). LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access, 6*, 1662-1669. doi:10.1109/ACCESS.2017.2779939

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks.* Paper presented at the Advances in neural information processing systems.

Li, P., Abdel-Aty, M., & Yuan, J. (2020). Real-time crash risk prediction on arterials based on LSTM-CNN. *Accident Analysis & Prevention, 135*, 105371. doi:https://doi.org/10.1016/j.aap.2019.105371

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. *Transportation Research Part C: Emerging Technologies, 55*, 444-459. doi:10.1016/j.trc.2015.03.015

Lin, T., Guo, T., & Aberer, K. (2017). *Hybrid neural networks for learning the trend in time series.* Paper presented at the Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence.

Liu, Y., Zheng, H., Feng, X., & Chen, Z. (2017, 11-13 Oct. 2017). *Short-term traffic flow prediction with Conv-LSTM.* Paper presented at the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP).

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors (Basel), 17*(4). doi:10.3390/s17040818

Ma, X., Zhang, J., Du, B., Ding, C., & Sun, L. (2019). Parallel Architecture of Convolutional Bi-Directional LSTM Neural Networks for Network-Wide Metro Ridership Prediction. *IEEE transactions on intelligent transportation systems, 20*(6), 2278-2288. doi:10.1109/TITS.2018.2867042

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research, 9*(Nov), 2579-2605.

Oh, J.-S., Oh, C., Ritchie Stephen, G., & Chang, M. (2005). Real-Time Estimation of Accident Likelihood for Safety Enhancement. *Journal of Transportation Engineering, 131*(5), 358-363. doi:10.1061/(ASCE)0733-947X(2005)131:5(358)

Olah, C. (2015). Understanding LSTM Networks. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*(Oct), 2825-2830.

Rahman, M. H., Abdel-Aty, M., Lee, J., & Rahman, M. S. (2019). Enhancing traffic safety at school zones by operation and engineering countermeasures: A microscopic simulation approach. *Simulation Modelling Practice and Theory, 94*, 334-348. doi:https://doi.org/10.1016/j.simpat.2019.04.001

Shi, Q., & Abdel-Aty, M. (2015). Big Data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. *Transportation Research Part C: Emerging Technologies, 58*, 380-394. doi:10.1016/j.trc.2015.02.022

Theofilatos, A. (2017). Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials. *Journal of Safety Research, 61*, 9-21. doi:https://doi.org/10.1016/j.jsr.2017.02.003

Tian, C., Ma, J., Zhang, C., & Zhan, P. (2018). A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. *Energies, 11*(12), 3493.

Tian, Y., & Pan, L. (2015, 19-21 Dec. 2015). *Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network.* Paper presented at the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity).

Wang, L., Abdel-Aty, M., Shi, Q., & Park, J. (2015). Real-time crash prediction for expressway weaving segments. *Transportation Research Part C: Emerging Technologies, 61*, 1-10. doi:10.1016/j.trc.2015.10.008

Wang, Z., Yan, W., & Oates, T. (2017, 14-19 May 2017). *Time series classification from scratch with deep neural networks: A strong baseline.* Paper presented at the 2017 International Joint Conference on Neural Networks (IJCNN).

Xu, C., Tarko, A. P., Wang, W., & Liu, P. (2013). Predicting crash likelihood and severity on freeways with real-time loop detector data. *Accident Analysis & Prevention, 57*, 30-39. doi:https://doi.org/10.1016/j.aap.2013.03.035

Yu, R., & Abdel-Aty, M. (2013). Utilizing support vector machine in real-time crash risk evaluation. *Accident Analysis & Prevention, 51*, 252-259. doi:https://doi.org/10.1016/j.aap.2012.11.027

Yu, R., & Abdel-Aty, M. (2014). Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data. *Safety Science, 63*, 50-56. doi:10.1016/j.ssci.2013.10.012

Yu, R., Wang, X., Yang, K., & Abdel-Aty, M. (2016). Crash risk analysis for Shanghai urban expressways: A Bayesian semi-parametric modeling approach. *Accident Analysis & Prevention, 95*, 495-502. doi:https://doi.org/10.1016/j.aap.2015.11.029

Yuan, J., & Abdel-Aty, M. (2018). Approach-level real-time crash risk analysis for signalized intersections. *Accident Analysis & Prevention, 119*, 274-289. doi:https://doi.org/10.1016/j.aap.2018.07.031

Yuan, J., Abdel-Aty, M., Gong, Y., & Cai, Q. (2019). Real-Time Crash Risk Prediction using Long Short-Term Memory Recurrent Neural Network. *Transportation Research Record, 2673*(4), 314-326. doi:10.1177/0361198119840611

Yuan, J., Abdel-Aty, M., Wang, L., Lee, J., Yu, R., & Wang, X. (2018). Utilizing bluetooth and adaptive signal control data for real-time safety analysis on urban arterials. *Transportation Research Part C: Emerging Technologies, 97*, 114-127. doi:https://doi.org/10.1016/j.trc.2018.10.009

Yue, L., Abdel-Aty, M., Wu, Y., & Wang, L. (2018). Assessment of the safety benefits of vehicles' advanced driver assistance, connectivity and low level automation systems. *Accident Analysis & Prevention, 117*, 55-64. doi:https://doi.org/10.1016/j.aap.2018.04.002

Zhang, J., Ma, X., Ding, C., Wang, Y., & Liu, J. (2018). *Forecasting subway demand in large-scale networks: a deep learning approach*. Retrieved from

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. *Transportation Research Record*, 0361198120912422. doi:10.1177/0361198120912422

Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems, 11*(2), 68-75. doi:10.1049/iet-its.2016.0208

Zheng, Z., Yang, Y., Liu, J., Dai, H., & Zhang, Y. (2019). Deep and Embedded Learning Approach for Traffic Flow Prediction in Urban Informatics. *IEEE Transactions On Intelligent Transportation Systems*, 1-13. doi:10.1109/TITS.2019.2909904

Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*.