Neurocomputing 423 (2021) 135–147

Contents lists available at ScienceDirect

Neurocomputing

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

Deep spatio-temporal graph convolutional network for traffic accident prediction

Le Yu, Bowen Du, Xiao Hu, Leilei Sun *, Liangzhe Han, Weifeng Lv

SKLSDE and BDBC Lab, Beihang University, No. 37 Xueyuan Road, Haidian District, Beijing 100083, PR China

ARTICLE INFO

Article history:
Received 21 November 2019
Revised 23 September 2020
Accepted 26 September 2020
Available online 6 October 2020

Keywords:
Traffic accident prediction
Spatio-temporal data
Graph convolutional network
Deep learning

ABSTRACT

Traffic accidents usually lead to severe human casualties and huge economic losses in real-world scenarios. Timely accurate prediction of traffic accidents has great potential to protect public safety and reduce economic losses. However, it is challenging to predict traffic accidents due to the complex causality of traffic accidents with multiple factors, including spatial correlations, temporal dynamic interactions and external influences in traffic-relevant heterogeneous data. To overcome the above issues, this paper proposes a novel Deep Spatio-Temporal Graph Convolutional Network, namely DSTGCN, to predict traffic accidents. The proposed model is composed of three components: the first component is the spatial learning layer which performs graph convolutional operations on spatial information to learn the correlations in space. The second component is the spatio-temporal learning layer which utilizes graph and standard convolutions to capture the dynamic variations in both spatial and temporal perspective. The third component is the embedding layer which aims to obtain meaningful and semantic representations of external information. To evaluate the proposed model, we collect large-scale real-world data, including accident records, citi-wide vehicle speeds, road networks, meteorological conditions, and Point-of-Interest distributions. Experimental results on real-world datasets demonstrate that DSTGCN outperforms both classical and state-of-the-art methods.

1. Introduction

In recent years, a large amount of traffic data have been collected due to the boom of vehicle usage and rapid development of intelligent transportation systems. Such massive traffic data arouse the interests of many scholars on designing data-driven methods to discover the hidden valuable information, which could not only help people choose more efficient ways for traveling, but also promote relevant departments to run cities better. By now, many urban problems, including traffic congestion and environmental pollution, have attracted much attention and some methods have been proposed to alleviate these two problems effectively [1–4]. While as another rarely studied urban problem, traffic accident prediction has not been solved with satisfying approaches yet, which could avoid huge damage on both lives and properties of human beings.

According to the global status report on road safety by World Health Organization in 2018, about 1.35 million people are killed every year because of traffic accidents.[1] In order to reduce the loss caused by traffic accidents, it is essential to estimate the probability of whether a traffic accident will happen or not in a specific region ahead of time. Accurate assessment of traffic accidents happening risk would instruct drivers to select a safer path or help urban traffic management departments make decisions and take emergency measures in advance.

However, it is a non-trivial endeavor to predict traffic accidents due to the following reasons. First, traffic accidents only occur nearby roads and the granularity of prediction should be specified in a road-level, which is more meaningful in real-world scenarios. Second, there are multiple factors which could be relevant to traffic accidents, such as driver behaviors, weather conditions, traffic flow and road structures. Although some researchers have studied the key factors causing traffic accidents [5,6], they did not take indirect factors into consideration and the complex mechanism of traffic accident is still unclear. Third, the occurrence of traffic accidents

* Corresponding author.
E-mail addresses: yule@buaa.edu.cn (L. Yu), dubowen@buaa.edu.cn (B. Du), xiaohu@buaa.edu.cn (X. Hu), leileisun@buaa.edu.cn (L. Sun), liangzhehan@buaa.edu.cn (L. Han), lwf@buaa.edu.cn (W. Lv).

[1] https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/.

https://doi.org/10.1016/j.neucom.2020.09.043
0925-2312/

is quite sparser than that of normal safe travels, resulting in a lack of positive samples for precise traffic accident prediction. The imbalance of samples enhances the difficulty in model training process significantly.

By now, many scholars have investigated the problem of traffic accident prediction, which could be formulated as a classification problem or a regression problem. On the one hand, a few scholars attempted to predict traffic accidents by classical machine learning techniques, including regression models [7], K-Nearest Neighbor (KNN) [8], Bayesian networks [9] and decision trees [10]. Most of them treated the prediction of traffic accidents as a classification problem, which aims to predict whether a traffic accident would happen in the future or not. They hardly considered the complicated factors aforementioned, and as a result, the performance on prediction accuracy was relatively unsatisfying. On the other hand, a few deep learning approaches have been applied in this field [11–13]. Instead of modeling predicting traffic accidents as a binary classification problem, Chen et al. [11] and Ren et al. [12] tried to estimate the risk of traffic accidents, while Yuan et al. [13] aimed to predict the number of traffic accidents at a given time and location. However, they either ignored the correlations of spatio-temporal information or just made prediction in a coarse-grained way by dividing studied areas into regular grids. Hence, a fine-grained traffic accident prediction method considering heterogeneous spatio-temporal information is eagerly required.

To cope with the above problems, in this paper, we propose a Deep Spatio-Temporal Graph Neural Network, namely DSTGCN, which aims to predict the risk of traffic accidents in the future for specific road segments. First, in order to predict traffic accidents in a road-level, we construct a graph according to fine-grained road structure in the studied area and make prediction based on the graph. Second, we collect large-scale heterogeneous data relevant to traffic accidents, including weather conditions, traffic flows, road structures, Point Of Interest (POI) distributions and traffic accident records. Then we design a deep spatio-temporal graph neural network to take these data as inputs simultaneously and discover spatial correlations and temporal dependencies of the mechanism causing traffic accidents. Moreover, a data sampling method is applied to deal with the problem of imbalanced samples. Finally, experimental results on real-world traffic data demonstrate that the proposed model achieves better performance than both the classical and state-of-the-art methods.

In summary, we highlight our contributions as follows:

- We investigate the problem of traffic accident prediction in road-level based on graph-structured data, which would be a more feasible and suitable way for studying traffic accidents in reality.
- A novel deep spatio-temporal graph convolutional neural network is proposed to consider both spatio-temporal relationships and external influences in massive heterogeneous data simultaneously for more accurate prediction.
- The proposed model is evaluated on real-world data and experimental results demonstrate the advantage of our approach compared to existing methods.

The rest of the paper is organized as follows: Section 2 reviews existing research relevant to this paper. Section 3 introduces the background knowledge, problem formalization and describes datasets to evaluate the model. Section 4 presents feature extraction process and the proposed model. Section 5 shows experimental results and discusses the effects of different features and model structures. Section 6 concludes the entire paper.

2. Related work

In this section, we review the existing literature related to our work, and also point out the limitations of previous studies.

2.1. Traffic accident prediction

With the significant advancements in machine learning, many traffic accident prediction methods based on such techniques have been provided. Chang and Chen [7] established the empirical relationship between traffic accidents and highway geometric variables by developing a CART model and a negative binomial regression model to predict accidents happened in highways. Lv et al. [8] selected features of traffic accidents based on Euclidean distance and completed the real-time highway traffic accident prediction by using KNN method with traffic data. Hossain and Muromachi [9] applied Bayesian Belief Net (BBN) to build real-time highway crash prediction model. Lin et al. [10] proposed a Frequent Pattern tree (FP-tree) based method to select variable features that are more likely to contribute to the traffic accident. Then they utilized KNN and Bayesian Network to predict traffic accidents by using selected features. They mainly applied classical machine learning methods to make the prediction by using hand-designed features extracted from traffic accident data. Nonetheless, traffic accidents are caused by many complex factors, including traffic flow, meteorological condition, road networks, etc. Without considering the spatio-temporal correlations and external information, the predicting accuracy and model performance are relatively unsatisfactory.

Recently, some scholars focus on the application of deep learning methods in traffic accident prediction field [11–13]. Chen et al. [11] utilized human mobility data and historical traffic accident records in Japan and built a Stack denoising Auto-Encoder (SdAE) model to infer real-time traffic accident risk. It was the first attempt to use a deep learning method to estimate traffic accident risk in a national scale. However, this work could only obtain a real-time accident risk map, which was not suitable for near future traffic accident prediction. Moreover, it only took human mobile phones' GPS data into consideration while many other factors were ignored. Ren et al. [12] exploited Long Short-Term Memory (LSTM) and Fully Connected (FC) network to predict traffic accident risk in Beijing in the future. They took several important factors into account, including weather conditions, speed of traffic flows and geographical positions. But the granularity of prediction could not be specified in the road level. Yuan et al. [13] proposed a Hetero-Convolutional Long Short-Term Memory (Hetero-ConvLSTM) model to predict traffic accident amounts in Iowa, which incorporated both spatial and temporal features, including time-invariant features, time-variant features and spatial-graph features. But this work made prediction in a coarse-grained way, which is constrained in grid cells spatially and daily prediction temporally. All the existing deep learning based traffic accident prediction methods are limited in estimating the traffic accident risk for regular grid cells, which is unreasonable in real-world situations since traffic accidents only occur nearby the road network. Although Yuan et al. [13] employed a road network mask layer to map model final outputs to road networks, the predicting process was still limited to grids level because the filter mask layer is just a simple AND operation.

2.2. Graph convolutional network

In recent years, Convolutional Neural Network (CNN) has shown great power in the field of computer vision, such as image classification [14], object detection [15], and image caption [16].

However, the data that CNNs could handle are constrained in regular format (e.g., grid-structure in Euclidean space), and CNNs are not suitable for graph data (e.g., graph-structure in non-Euclidean space). By now, there is a great amount of research on Graph Neural Network (GNN) with the purpose to utilize graph data better, which could model the rich relationships of nodes in a more flexible way. Zhou et al. [17] and Wu et al. [18] conducted systematic survey and summarized existing methods of GNNs under a unified framework. Recently, there is an increasing interest in Graph Convolutional Network (GCN) because of its efficiency and convenience. Generally, GCNs could be categorized as spectral-based and spatial-based methods.

Spectral-based methods, which are founded on graph signal processing theory [19], work with a spectral representation of graphs. In order to investigate topological structures of a graph (e.g., the connectivity), such methods convert the original graph into a algebraic form by the Fourier transform and update node's features in the transformed space.

Spatial-based methods define convolution operations directly on the graph and convolve the representations of a central node and its neighbors to obtain the new representation for the central node, which is inspired by the convolutional operation in CNNs on images [20]. Spatial-based methods maintain the local invariance of CNNs because the learning process could be seen as the information propagation among nodes based on edges, which is not affected by the operation orders with nodes. GCN has been applied successfully in a wide range of real-world applications. For example, in timely traffic forecasting, Yu et al. [21] built Spatio-Temporal Graph Convolutional Networks (STGCN), which provided faster training speed with fewer parameters and higher accuracy. In the field of healthcare, Li et al. [22] proposed a spatio-temporal graph convolution (STGC) approach for skeleton-based action recognition. Due to the advantages of GCN, we predict traffic accidents under a graph-based framework.

2.3. Spatio-temporal neural network

Since the concept of deep learning has been proposed [23], many deep spatio-temporal models have been applied in several fields, such as crowd flows prediction [24], traffic demand prediction [25], air quality inference [26,27] and crime rate forecasting [28]. Zhang et al. [24] proposed a deep model based on spatio-temporal data, namely ST-ResNet, to predict the inflow and outflow of crowds in each region of a city. Lv et al. [25] took advantage of both Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) to raise Look-up Convolution Recurrent Neural Network (LC-RNN) model to predicting traffic speed accurately in road segment granularity. Cheng et al. [26] utilized both the information from monitoring stations and urban data that are closely related to air quality and built an Attentional Deep Air quality Inference Network (ADAIN) model for urban air quality inference. Yi et al. [27] proposed a deep neural network-based model (DeepAir), which consists of a spatial transformation component and a deep distributed fusion network, to predict the air quality of next two days for each monitoring station. Wang et al. [28] adopted ST-ResNet to forecast crime distributions over Los Angeles area.

3. Data preparation and problem formalization

In this section, we first introduce and analyze the collected heterogeneous spatio-temporal data and then we formalize the studied problem.

3.1. Data preparation

In order to predict traffic accidents, in this study, large-scale and heterogeneous data related to traffic accidents are collected from Beijing. The details of all the data are shown as follows:

Traffic Accident Data. Traffic accident records of Beijing are collected during the period from 2018/08/01 to 2018/10/31 hourly, which consist of the timestamps and locations of traffic accidents. As shown in Fig. 1, the accident occurrence location is strongly related to the urban road network, which indicates existing works for traffic accidents based on CNNs ignoring the spatial adjacency between road segments are not reasonable [11–13].

Taxi GPS Data. Beijing taxi GPS data are obtained from 2018/08/01 to 2018/10/31 which were recorded every five minutes. In addition to timestamps and locations, taxi GPS data also contain the speed of each taxi.

POI Data. A POI is a specific point location offering a specific service for someone, such as a shopping center, a factory or a residential district. The POI dataset we collected contains 362,028 POIs in Beijing, including information of POI's name, location and category. Table 1 shows top 10 types of POIs in Beijing.

Meteorological Data. We crawl meteorological data from wunderground.[2] The dataset is collected by Tianning Temple Station (39.86°N, 116.28°E) during the period between 2018/08/01 and 2018/10/31, which is recorded hourly. This dataset contains the meteorological information, such as temperature, weather types, and so on. Fig. 2(a) shows the relationship between traffic accident frequency and temperature, and Fig. 2(b) presents how different weather conditions influence traffic accident frequency. From Fig. 2, it could be concluded that higher temperature and more severe weather condition are easier to cause frequent traffic accidents.

Road Network Data. The road network data of Beijing are also used. The data contain basic information of Beijing road network, such as name, set of points that each road contains, road intersection points and road lengths. More details of the datasets are shown in Table 2.

3.2. Problem formalization

We define an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ representing the urban road network, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denotes the collection of N road segments and $\mathcal{E}$ represents for the set of edges which indicates the pairwise connectivity between road segments. $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the traffic network $\mathcal{G}$. $A_{i,j}$ is set to 1 if road segments v_i and v_j are connected with each other, and 0 otherwise.

The goal of traffic accident prediction is to predict the next-period accident risk at a specified road segment v_i according to historical records. Table 3 lists major notations of features used in this paper. More details of traffic network generation and feature extraction will be introduced in Section 4.1.

For road segment v_i, the next-period traffic accident risk at time slot $T+1$ could be estimated by two steps as follows,

- *Feature extraction*, where the impact factors of traffic accident are classified into three categories: spatial features $\boldsymbol{x}_i^{spatial} = \left[\boldsymbol{x}_i^S; \boldsymbol{x}_i^P\right] \in \mathbb{R}^{S+P}$ indicating the influence factors surrounding v_i, temporal features $\boldsymbol{x}_i^{temporal} = \left\{x_i^{V,t}\right\}_{t=1}^{T} \in \mathbb{R}^{1 \times T}$ indicating the historical traffic condition of v_i and external features $\boldsymbol{x}^{external} = \left[\boldsymbol{x}^{M,T}; \boldsymbol{x}^{C,T+1}\right] \in \mathbb{R}^{M+C}$ indicating the environmental factors.

[2] https://www.wunderground.com/.

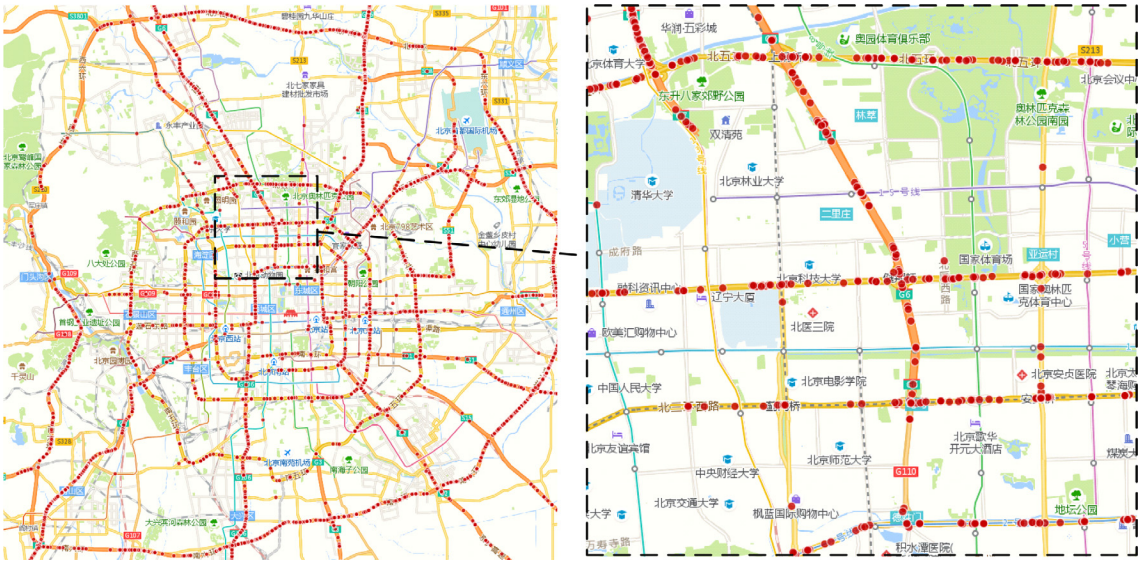


Fig. 1. The spatial distribution of traffic accidents.

Table 1
Top 10 types of POIs.

Category	Examples	Frequency
Shopping	Shopping malls	67,781
Food	Restaurants, dining halls	51,612
Working place	Companies, office buildings	50,125
Daily service	Post offices, repairs, clubs	43,789
Home	Residential buildings	26,700
Government	Police stations, courts	17,901
Education	Schools, training centers	17,271
Public facility	Toilets and telephone booth	16,485
Transportation	Airports, transit centers	15,933
Healthcare	Hospitals, pharmacies	10,770

- *Model prediction*, which estimates the probability of traffic accident happening on road segment v_i at time slot $T+1$ according to its extracted features $\boldsymbol{x}_i = \left\{\boldsymbol{x}_i^{spatial}, \boldsymbol{x}_i^{temporal}, \boldsymbol{x}^{external}\right\}$ and its neighbors' features $\boldsymbol{X}^{neighbors} = \{\boldsymbol{x}_j : j \in \mathcal{N}(v_i)\}$ with trainable weights $\boldsymbol{W}$,

$$\hat{y}_i = f\left(\boldsymbol{x}_i, \boldsymbol{X}^{neighbors}; \boldsymbol{W}\right),$$

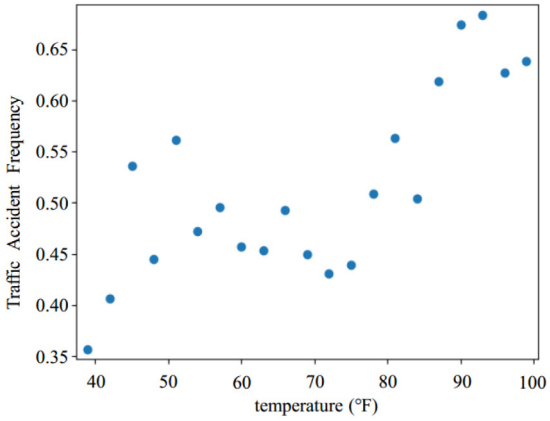
where $\mathcal{N}(v_i)$ stands for k-hop neighbors of v_i, which is a set of road segments that are reachable from v_i with k or fewer hops (i.e., a path consists of k or fewer edges in $\mathcal{E}$) in the graph $\mathcal{G}$. The model weights $\boldsymbol{W}$ are trained by fitting the previous traffic accident events by minimizing the following equation,

$$\mathcal{L} = -\frac{1}{\zeta}\sum_{i=1}^{\zeta}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)) + \lambda\|\boldsymbol{W}\|^2, \quad (1)$$

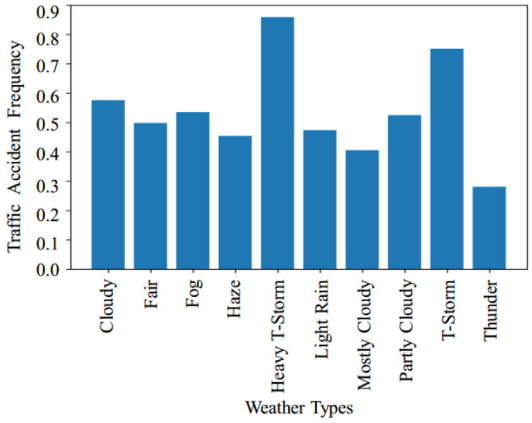
where ζ is the number of training samples, y_i is the ground truth of traffic accidents risk of sample i and λ is the $L2$ regularization parameter.

4. Methodology

This section first presents the process of feature extraction, including generating road network and extracting different categories of features. Then the framework of our proposed model is



(a) Traffic accident frequency vs. temperature.



(b) Traffic accident frequency vs. weather types.

Fig. 2. Traffic accident frequency varies with weather conditions, such as temperature and weather types.

Table 2
Statistics of the datasets.

Data	Statistics	
Traffic Accidents	Records	14,874
POIs	Categories	20
	# Number	362,028
Taxi GPS	Time	2018/08/01 to 2018/10/31
	# Number	About 54,000,000 a day
Meteorological Data	Time	2018/08/01 to 2018/10/31
	# Number	2,208
Road Networks	# Number of side roads	109,973

Table 3
Notations and explanations.

Notations	Explanations
$\boldsymbol{x}_i^S$	Static features of road segment v_i, e.g., length, direction, position, $\boldsymbol{x}_i^S \in \mathbb{R}^S$.
$\boldsymbol{x}_i^P$	POI distributions around road segment $v_i, \boldsymbol{x}_i^P \in \mathbb{R}^P$.
$x_i^{V,t}$	Traffic speed of road segment v_i at time slot $t, x_i^{V,t} \in \mathbb{R}$.
$\boldsymbol{x}^{M,t}$	Meteorological conditions at time slot $t, \boldsymbol{x}^{M,t} \in \mathbb{R}^M$.
$\boldsymbol{x}^{C,t}$	Calendar representations at time slot t, e.g., day of the week, $\boldsymbol{x}^{C,t} \in \mathbb{R}^C$.
y_i^t	Whether there is an accident on road segment v_i at time slot $t, y_i^t \in \{0,1\}$

introduced. Finally, details of the proposed model are discussed step by step.

4.1. Feature extraction

As aforementioned, the impact factors of traffic accident fall into three categories: spatial features, temporal features and external features. In this section, we introduce the process of road network generation and extraction of each features, respectively.

4.1.1. Road network graph generation

The original road network $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ data used in this paper is provided by [29], consisting of a set of nodes as intersections and a set of edges as road segments that connect the nodes, and the length of edges are calculated. We aim to predict traffic accidents in road-level, so roads should be treated as nodes in the graph. Herein, we construct the node set $\mathcal{V}$ by the provided edges $\mathcal{E}'$, which means that the i-th node $v_i \in \mathcal{V}$ corresponds to the i-th edge $e_i' \in \mathcal{E}'$. Then we construct edge set $\mathcal{E}$ by adding an edge between v_i and v_j into $\mathcal{E}$ when road segments $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ are connected by an intersection in $\mathcal{V}'$:

$$\mathcal{E} = \left\{ (v_i, v_j) : v_i, v_j \in \mathcal{V} \wedge \left(\exists v' \in \mathcal{V}' \quad \text{s.t.} \quad e_i' \wedge e_j' = v' \right) \right\}.$$

4.1.2. Extraction of spatial features

The spatial features $\boldsymbol{x}_i^{spatial} = [\boldsymbol{x}_i^S; \boldsymbol{x}_i^P]$ consists of road structure features $\boldsymbol{x}_i^S$ and POI distribution $\boldsymbol{x}_i^P$, depicting the local spatial features of each road segment v_i, which have direct or indirect impacts on the risk of traffic accidents.

The feature $\boldsymbol{x}_i^S$ of each road segment v_i represents the attribute of road structure which is assumed to influence the traffic accident risk directly as roads with more complicated conditions tend to occur traffic accidents more easily. We average the positions of points belonging to each road segment as the road's geographical position. And then, features related to the road structure, including road length and the number of containing points, are extracted for each road segment.

POI distribution $\boldsymbol{x}_i^P$ of each road segment v_i depicts its local surroundings, which are assumed to impact traffic accident risk in an indirect way. For example, roads surrounded by recreation centers or a parking lot may at a higher traffic accident risk than those nearby quiet parks. As POI distribution well captures the characteristics of road segments, we leverage POI data and extract POI features for traffic accident prediction. A set of 20 types of POIs $\mathbb{D}^P$ are taken into consideration. The number of each POI category nearby road segment v_i is calculated as follows,

$$x_{i,j}^P = \left| \{ p : p \in \mathbb{D}^{P,j} \wedge dist(v_i, p) \leqslant d \} \right|, \text{ for } 1 \leqslant j \leqslant 20,$$

where $dist(v_i, p)$ denotes the distance between a POI p and a road segment v_i, $|\cdot|$ denotes the cardinality of a set and $\mathbb{D}^{P,j}$ represents the set containing POIs of type j in Beijing. d is a hyperparameter to control the local horizon of each road segment.

4.1.3. Extraction of temporal features

The temporal feature $\boldsymbol{x}_i^{temporal} = \left\{ x_i^{V,t} \right\}_{t=1}^{T}$ impacts the risk of traffic accidents temporally, as it could reflect the historical traffic condition of each road segment v_i.

Intuitively, the speed of traffic flow is closely relevant to the occurrence probability of traffic accident. Therefore, we calculate the average traffic speed of each road segment v_i in each time slot t denoted by $x_i^{V,t}$. Since the taxi data have a large scale, and it is expensive and unfeasible to traverse for each road segment step by step. So we first partition the studied area into equal-sized grids and calculate each grid's traffic speed in each time slot by averaging the taxi speed inside each grid. Then we allocate each road to the grid it belongs to and finally assign the grid's traffic speed to the road as its own traffic flow speed feature. Each grid is a $d \times d$ square and the traffic speed of each road segment v_i is determined by its located grid. And d is the hyperparameter aforementioned for the adjustment of each road segment's scope. We use the speed in the last 24 h ahead the time of prediction and each hour corresponds to a dimension of the temporal feature.

4.1.4. Extraction of external features

In addition to the spatial and temporal features represented by $\boldsymbol{x}_i^{spatial}$ and $\boldsymbol{x}_i^{temporal}$ separately, there are external factors $\boldsymbol{x}^{external} = [\boldsymbol{x}^{M,T}; \boldsymbol{x}^{C,T+1}]$ that influence the risk of traffic accident. The external features considered in this paper are comprised of meteorological features $\boldsymbol{x}^M$ and calendar features $\boldsymbol{x}^C$. We do not use the subscript i to discriminate different roads, because all the roads have identified external features at the same timestamp.

Empirically, meteorological features ($\boldsymbol{x}^M$) may influence traffic accident easily since severe weather (e.g., snow, fog, rain) could increase the risk of traffic accident, see Fig. 2. Hence, eight meteorological factors are considered, including weather type, weather temperature, dew point, humidity, pressure, wind speed, wind direction and apparent temperature. Among these factors, weather type and wind direction are categorical attributes with 14 and 18 values respectively, while the others are numerical attributes. We utilize one-hot encoding to denote weather type and wind direction. Let $\boldsymbol{x}^{M,t}$ denote the meteorological features at time slot t shared by all road segments. Since the meteorological situation at future time $T+1$ are not available until the time has passed, we adopt the meteorological characteristics at time T to approximate the future meteorological features in this work.

Calendar Features ($\boldsymbol{x}^C$) (e.g., month, day of the week, weekday or weekend, hour) may also have an effect on the frequency of traffic accidents, since it reflects the periodicity of traffic behavior to a certain extent. Let $\boldsymbol{x}^{C,t}$ represent the calendar information at time

slot t containing five features, which includes month, day, day of the week, hour and whether it is a weekend.

Finally, we could generate 89 features, with 22 spatial features (2 road structure features and 20 POI features), 24 temporal features (24 traffic flow speed features) and 43 external features (38 meteorological features and 5 calendar features) for each road segment.

4.2. Framework of the proposed model

Fig. 3 illustrates the framework of our proposed approach. The traffic network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ is generated using road network data. Then features from heterogeneous data (i.e., road networks, taxi GPS data, meteorological data, traffic accidents records, POIs) are extracted for all road segments. As different features have different effects on the happening probability of future traffic accidents, we divide extracted features into three categories: spatial features that could reflect the spatial local characteristics of road segments; temporal features that record the historical traffic conditions of each road segment; external features that describe external influences.

For each training sample, DSTGCN first deals with different types of features based on three components, the Spatial Convolution Layer, the Spatio-Temporal Convolution Layer and the Embedding Layer respectively. Then the processed hidden features are concatenated into a compact representation and fed into a Fully Connected (FC) network to learn interactions among different features and predict the risk of next-period traffic accidents. Finally, in order to evaluate the proposed model, experiments are conducted on real-world datasets. The proposed DSTGCN is compared with both classical and state-of-the-art baselines, and the effects of different features and model structures are investigated.

4.3. Proposed model for traffic accident prediction

The structure of the proposed model is shown in Fig. 4, which consists of three basic modules: the spatial convolution layer, temporal convolution layer and FC layer. Next, we will present basic modules and three components for modeling spatial features, temporal features and external features respectively.

4.3.1. Basic modules

We construct our proposed model by three types of basic modules. Since the FC layer is simply a linear transformation followed by an activation function, the spatial convolution layer and temporal convolution layer will be elaborated as below:

Spatial Convolution Layer: We first review the basic concepts in graph convolution operations. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$, where $\mathcal{V}$ is the set of N nodes and $\mathcal{E}$ is the set of edges, graph signal $\boldsymbol{h}_i \in \mathbb{R}^F$ on $\mathcal{G}$ is defined by F features of node i. Graph convolution on l-th layer is introduced by [30] and can be described as below:

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\boldsymbol{b}^l + \sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} \boldsymbol{h}_j^{(l)} \boldsymbol{W}^l\right), \tag{2}$$

where $\mathcal{N}(i)$ is the neighbor set of node i. c_{ij} is equal to the product of the square root of node degrees $\sqrt{|\mathcal{N}(i)|}$. σ is an activation function, and $\boldsymbol{W}^l \in \mathbb{R}^{F \times F'}$ and $\boldsymbol{b}^l \in \mathbb{R}^{F'}$ are trainable weights and biases in the l-th layer. The graph convolution in the l-th layer aggregates graph signals from neighbors of node v_i after a shared affine transformation, and produces new representation for each node.

Moving the convolution kernel $\Theta = \left\{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\right\}$ over the graph, the convolution operation over the whole graph with input signal and output signal are separately $\boldsymbol{H}^{(l)} = \left(\boldsymbol{h}_1^{(l)}, \boldsymbol{h}_2^{(l)}, \ldots, \boldsymbol{h}_N^{(l)}\right) \in \mathbb{R}^{N \times F}$ and $\boldsymbol{H}^{(l+1)} = \left(\boldsymbol{h}_1^{(l+1)}, \boldsymbol{h}_2^{(l+1)}, \ldots, \boldsymbol{h}_N^{(l+1)}\right) \in \mathbb{R}^{N \times F'}$ is denoted by:

$$\boldsymbol{H}^{(l+1)} = \boldsymbol{H}^{(l)} *_{\mathcal{G}} \Theta, \tag{3}$$

where $\boldsymbol{h}_i^{(l+1)}$ are calculated with the shared parameters $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ and $N = |\mathcal{V}|$. $*_{\mathcal{G}}$ denotes the symbol of graph convolution.

Based on the above concepts, we model the spatial relationships between road segments with topological structure and employ graph convolutions to capture the spatial correlations. To propagate spatial information, the spatial convolution layer consists of a graph convolution, which aggregates spatial information from road segment v_i and its neighborhood. A batch normalization [31] is used to increase the robustness of our model in initialization and accelerate the training speed, and a *ReLU* activation function for capturing non-linearity correlation. Specifically, for the l-th layer, assuming its input and output are $\boldsymbol{H}^{(l)} \in \mathbb{R}^{N \times F}$ and $\boldsymbol{H}^{(l+1)} \in \mathbb{R}^{N \times F'}$ respectively, the spatial graph convolution is calculated as follows,

$$\boldsymbol{H}^{(l+1)} = ReLU\left(BN\left(\boldsymbol{H}^{(l)} *_{\mathcal{G}} \Theta\right)\right), \tag{4}$$

where *BN* is batch normalization, *ReLU* is the activation function and Θ denotes trainable parameters.

We also extend the definition of graph signal to the temporal dimension. Formally, considering the input and output signals corresponding road segment v_i are $\mathcal{H}_i^{(l)} = \left(\boldsymbol{h}_{i,1}^{(l)}, \ldots, \boldsymbol{h}_{i,T}^{(l)}\right)$ and $\mathcal{H}_i^{(l+1)} = \left(\boldsymbol{h}_{i,1}^{(l+1)}, \ldots, \boldsymbol{h}_{i,T}^{(l+1)}\right)$, the convolution kernel $\Theta = \left\{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\right\}$ is shared both spatially and temporally:

$$\boldsymbol{h}_{i,t}^{(l+1)} = \sigma\left(\boldsymbol{b}^{(l)} + \sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} \boldsymbol{h}_{j,t}^{(l)} \boldsymbol{W}^{(l)}\right). \tag{5}$$

Finally, the output of road v_i in the l-th spatial convolution layer could be denoted as

$$\mathcal{H}_i^{(l+1)} = ReLU\left(BN\left(\mathcal{H}_i^{(l)} *_{\mathcal{G}} \Theta\right)\right). \tag{6}$$

Temporal Convolution Layer: While the graph convolution operations capture neighboring information for each node on the graph in spatial dimension, a standard convolution layer in temporal dimension is further adopted to update the signal of a node by merging the information in consecutive time slots.

Denoting the input signals of road v_i of l-th temporal convolution layer is $\mathcal{H}_i^{(l)} \in \mathbb{R}^{F \times T}$, the j-th channel of output signal $\mathcal{H}_i^{(l+1)} \in \mathbb{R}^{F \times T'}$ can be precisely calculated by,

$$\mathcal{H}_{i,j}^{(l+1)} = \sigma\left(\boldsymbol{b}_f^{(l)} + \sum_{k=0}^{F} \boldsymbol{W}_{f,k}^{(l)} * \mathcal{H}_{i,k}^{(l)}\right), \text{ for } 1 \leqslant f \leqslant F, \tag{7}$$

where $*$ is the valid cross-correlation operator. $\mathcal{H}_{i,k}^{(l)}$ represents the input signal's k-th channel at the l-th layer. The dimension of the convolution kernel size is 3x1, with stride of 1 and zero-padding of 1. Such a convolution kernel could keep the dimension of feature in output signal as the same as that in input signal (i.e., $T' = T$).

4.3.2. Fusion of multi-perspective features

We denote the spatial features as $\boldsymbol{X}^{spatial} = \left\{\boldsymbol{x}_1^{spatial}, \ldots, \boldsymbol{x}_N^{spatial}\right\}$, temporal features as $\boldsymbol{X}^{temporal} = \left\{\boldsymbol{x}_1^{temporal}, \ldots, \boldsymbol{x}_N^{temporal}\right\}$ and external features as $\boldsymbol{x}^{external}$ respectively. The following three components are constructed to take in the multi-perspective features and provide the risk of next-period traffic accident.

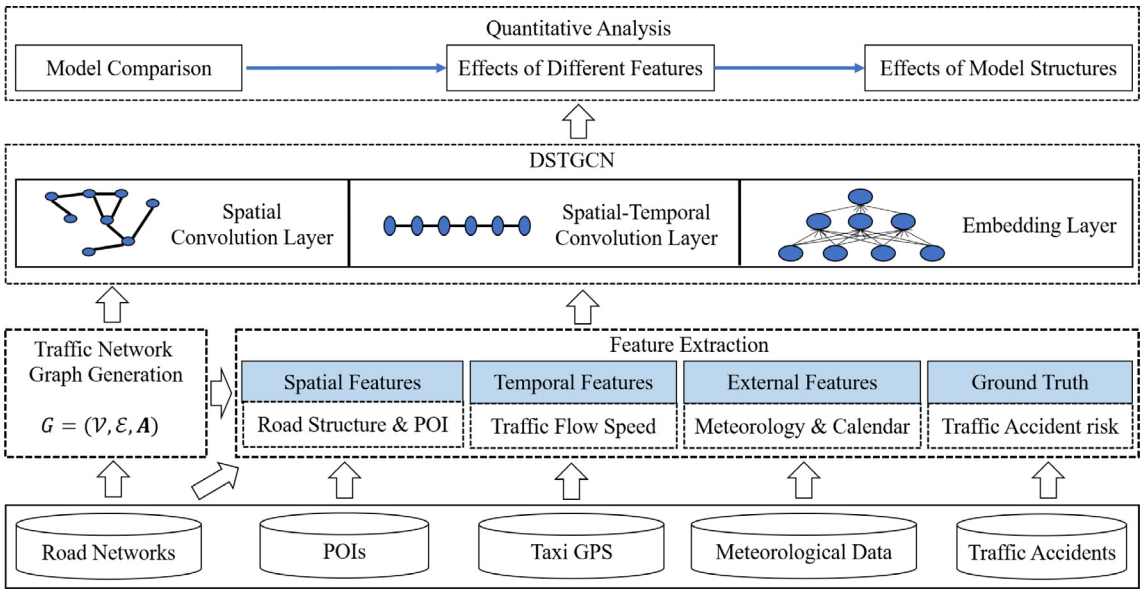


Fig. 3. Framework of the proposed approach.

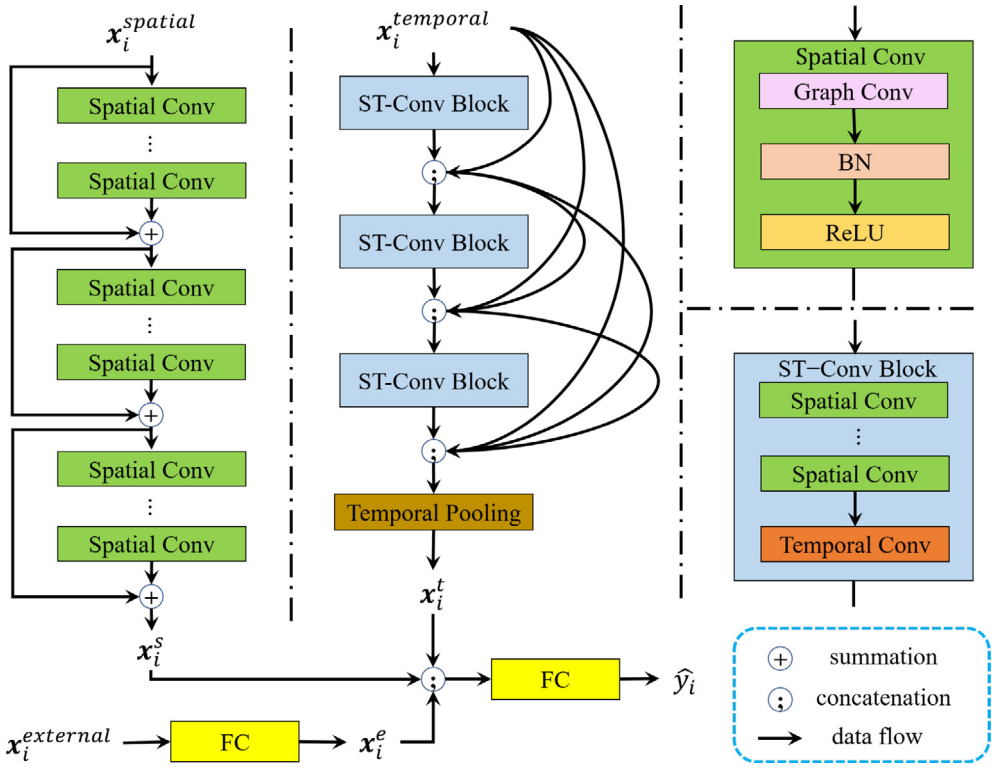


Fig. 4. The structure of the proposed model. Spatial Conv: Spatial Convolution Layer; Temporal Conv: Temporal Convolution Layer; FC: Fully Connected Layer.

Spatial Component consists of several spatial convolution blocks. The dimension of spatial features $\boldsymbol{X}^{spatial}$ is slightly reduced by a FC layer and then we feed the features into stacked spatial convolution blocks. A spatial convolution block is composed of several spatial convolution layers. In order to cope with the problem of vanishing/exploding gradients, the residual learning framework [32] is adopted. After the stacked spatial convolution blocks, the learned representation for spatial information is denoted by $\boldsymbol{X}^s \in \mathbb{R}^{N \times F_s}$, where F_s denotes the dimension of the output of spatial component. To estimate the risk of traffic accident on road segment v_i, we select its corresponding representation and denote it by $\boldsymbol{x}_i^s = \boldsymbol{X}_i^s \in \mathbb{R}^{F_s}$.

Temporal Component consists of several spatio-temporal convolution blocks. Each of them is composed of a series of stacked spatial convolution layers and a temporal convolution layer. The stacked spatial convolution layers are the same as previously mentioned spatial component. In order to alleviate the vanishing-gradient problem and strengthen feature propagation, we organize the spatio-temporal convolution blocks as densely connected as proposed by [33]. The blocks take in $\boldsymbol{X}^{temporal}$ as input and provide

a representation $\boldsymbol{X}^t \in \mathbb{R}^{N\times F_t\times T}$ of learned hidden temporal information, where F_t denotes the dimension of the output of spatio-temporal convolution blocks. To aggregate the temporal information into a compact representation, an average pooling method is adopted. Specifically, for road segment v_i, its temporal representation is calculated by,

$$\boldsymbol{x}_i^t = \frac{1}{T}\sum_{t'=1}^{T}\boldsymbol{X}_{i,:,t'}^t \in \mathbb{R}^{F_t}. \tag{8}$$

External Component: there may exist some noises in extracted high-dimension external features. In order to remove useless information and learn meaningful representations of external features, we employ an embedding layer in the external component to learn dense representations. Essentially, the embedding layer consists of several stacked FC layers, each FC layer could slightly reduce the dimension of original features. For road v_i, we feed the external feature $\boldsymbol{x}^{external}$ into an embedding layer and let $\boldsymbol{x}_i^e \in \mathbb{R}^{F_e}$ denote the embedded representation of external features, where F_e denotes the dimension of the embedding layer's output.

After getting outputs from the three components, we fuse the spatial, temporal and external information together by a concatenation operation into a fused representation. Then we feed the fused representation into a output layer to predict the risk of next-period traffic accident as follows:

$$\hat{y}_i = sigmoid\left(\boldsymbol{w}_o^\top \cdot \left[\boldsymbol{x}_i^s;\boldsymbol{x}_i^t;\boldsymbol{x}_i^e\right] + b_o\right), \tag{9}$$

where $\boldsymbol{w}_o \in \mathbb{R}^{F_s+F_t+F_e}$ and $b_o \in \mathbb{R}$ are trainable parameters in the output layer, $\cdot$ denotes the dot product, and ; denotes the concatenation of features.

5. Experiments

In this section, we evaluate the effectiveness of the proposed model by comparing with several existing methods through a series of experiments. Then we conduct ablation experiments to investigate the effects of different features and model structures.

5.1. Experimental setup

We first present an undersampling method to deal with the sparseness of samples. Then we introduce the model configurations in experiments. Finally, we discuss the evaluation metrics and baselines to be compared with the proposed model.

Undersampling Method. The occurrence of traffic accidents is quite sparser which means only a small proportion of roads have happened traffic accidents at a specific time. If we predict the risk of each road on the whole constructed graph of roads directly, the model would tend to provide all-zero results because of the sparsity of positive samples and achieve unsatisfactory performance. In order to solve the problem of sample sparseness, we adopt a negative sample undersampling method. For each traffic accident record, we first identify the road where the accident happened and then build a road network containing the road and its k-hop neighbors according to the road networks data. After that, we extract spatial, temporal and external features of the road and its k-hop neighbors. Following the aforementioned steps, we could obtain a graph containing the road and its k-hop neighbors and their features and treat it as a positive sample. We continue to perform the process until all traffic accidents are considered and finally get the collection of positive samples. After we randomly select a road where no traffic accidents happened during the specific period and extract information as aforementioned steps to generate a negative sample. Finally, the undersampling process would be finished when the number of negative samples with is equal to that of positive samples. Given the extracted spatial, temporal and external features of the target road segment and that of its k-hop neighbors, our model predicts whether there would be a traffic accident in the target road segment. We mark its ground truth with 1 if accidents have happened there, otherwise 0.

Baselines. We compare the proposed model with several existing methods, including Logistic Regression (LR) [34], Least Absolute Shrinkage and Selection Operator (LASSO) [35], Support Vector Machine (SVM) [36], Decision Tree (DT) [37], Stack Denoising Autoencoder (SdAE) [11] and Traffic Accident Risk Prediction Method based on LSTM (TARPML) [12]. In this paper, LR, LASSO, SVM and DT are treated as classical machine learning models. SdAE and TARPML are classified into state-of-the-art models. Since the concepts of LR, LASSO, SVM and DT are widely known, we refer the interested readers to [34–37] for more details. For state-of-the-arts, SdAE stacks denoising autoencoders to form deep networks and learn hierarchical feature representations of the inputs. Then the learned features are fed into a logistic regression layer to predict the risk of traffic accidents. TARPML takes in both static and dynamic information of factors influencing traffic accidents. FC layers are adopted to deal with static information and the LSTM unit is utilized to capture the dynamic information.

In fact, a generated training sample contains the spatial, temporal and external features of the road needs to be predicted and its k-hop neighbors, which is organized in a graph structure. Since the baselines could not take the graph-structure data with topological information as inputs, we process the topology structure of the graph as follows and then feed the non-topology data into the baselines. Specifically, for spatial and temporal features, we average the corresponding information of the predicted road and its k-hop neighbors and thus we could obtain two vectors of spatial and temporal features respectively. For external features which are non-topology and shared by roads in the studied area, we just need to maintain the vectorized representation. Finally, we concatenate the three obtained vectorized representations into a compact vector which could be fed into non-graph baselines. It is worth noting that we do not take the deep learning model in [13] into consideration as their model mainly relies on standard convolutional operations for regular grids-based data, which is not suitable for our experiments.

Evaluation Metric. Root Mean Squared Error (*RMSE*), Pearson's Correlation Coefficient (*PCC*), *Precision*, *Recall*, $F1-Score$, Area Under the Curve (*AUC*) are adopted to measure the performance of different methods as follows:

- *RMSE* is widely used in measuring the error of regression methods, which is defined as follows:

$$RMSE = \sqrt{\frac{1}{\xi}\sum_{i=1}^{\xi}\left(\hat{y}_i^{T+1} - y_i^{T+1}\right)^2},$$

  where $\hat{y}_i^{T+1}$ and y_i^{T+1} mean the prediction value and ground truth of the i_{th} sample at time interval $T+1$, and ξ stands for the total number of samples.
- *PCC* is used to measure the linear correlation between two variables. Its value range is from -1 to 1, where -1, 1 and 0 indicate total negative linear correlation, total positive linear correlation and no linear correlation respectively:

$$PCC = \frac{\sum_{i=1}^{\xi}\left(\hat{y}_i^{T+1} - \overline{\hat{y}^{T+1}}\right)\left(y_i^{T+1} - \overline{y^{T+1}}\right)}{\sqrt{\sum_{i=1}^{\xi}\left(\hat{y}_i^{T+1} - \overline{\hat{y}^{T+1}}\right)^2}\sqrt{\sum_{i=1}^{\xi}\left(y_i^{T+1} - \overline{y^{T+1}}\right)^2}},$$

  where $\overline{\hat{y}^{T+1}}$ and $\overline{y^{T+1}}$ are the average values of all predicted risks and ground truths.

- *Precision* is the ratio of correctly predicted positive observations to the all observations in predicted positive class, which is calculated by,

$$Precision = \frac{TP}{TP + FP},$$

- *Recall* is the ratio of correctly predicted positive observations to the all observations in actual positive class, which is defined as:

$$Recall = \frac{TP}{TP + FN},$$

- $F1 - Score$ represents the weighted average of *Precision* and *Recall*. $F1 - Score$ takes both false positives and false negatives into account which is usually more useful than *Precision* and *Recall*,

$$F1 - Score = \frac{2 * TP}{2 * TP + FN + FP},$$

where TP, TN, FP and FN stand for true positive, true negative, false positive and false negative respectively.

- *AUC* is one of the most important evaluation metrics for estimating a classification model's performance. It reflects the ability of a model in distinguishing between classes.

A model is regarded as a better model if it has lower *RMSE*, higher $PCC, Precision, Recall, F1 - Score$ and *AUC*. Since $F1 - Score$ is a more comprehensive metric, we utilize the metric to select better models. The model which achieves the highest $F1 - Score$ on the validation set is selected as the best model for testing. Note that our model aims to provide a probability of next period traffic accident which varies from 0 to 1 and could be treated as a regression task, and metrics like RMSE and PCC could be calculated directly. Moreover, if we choose a threshold to project the road with predicted risk to a positive or negative category, we could then use some metrics for classification tasks to validate the model performance. Hence, in this paper, we choose the threshold as 0.5 and if a road whose predicted risk is higher than 0.5, then it is treated as a positive sample, otherwise a negative sample. So we use not only regression metrics, but also classification metrics to make more comprehensive comparisons.

Implementation Details. According to the statistics, there are 1234251, 178414 and 350851 nodes involved in the training, validation and testing data respectively. To calculate the traffic flow speed, the value of d is set to 222 meters as one longitude or latitude degree corresponds to about 111 km, so 222 meters is roughly equal to 0.002 longitude or latitude degree. We conduct experiments with setting the value of k to 5, 10, 15 and 20, and the model performs best when 10-hop neighbors are considered. So we set k to 10 in the experiments, which means we consider the influence of 10-hop neighbors of each road. 70% of the data is selected as the training set for model training and 10% of the data is chosen as the validation set which is used to improve the generalization of our model and prevent the model from over-fitting. The remaining 20% is chosen as the set for testing. We choose the hold-out method instead of the cross-validation because the dataset is relative large. The cross-validation needs multiple train-test splits and would take more time and computational power to run than the hold-out method. To make a fair comparison, we split the training set, validation set and testing set using the same rule for all the models. We also normalize the input data by utilizing Z-score.

Table 4 shows the grid search of hyperparameters for all the methods. For the proposed DSTGCN, the first two spatial convolutional blocks both consist of four spatial convolution layers with skip-connections. The last block is made up of five spatial convolution layers to reduce the feature dimension. Each spatio-temporal convolutional blocks consists of five spatial convolution layers and a temporal convolution layer, and the blocks are connected with dense connections. The batch size is set to 64. The hidden dimensions of spatial, temporal and external features are set to 10, 20 and 10 respectively. Adam [38] is adopted as the optimizer in the experiments. During the training process, the *BN* layer keeps running estimations of its computed mean and variance, and its parameters are fixed and used for normalization for validation and testing. The model is implemented with the PyTorch[3] framework. We make the code publicly available on GitHub platform.[4]

Table 4
Details of the grid search in each method. The bold font denotes the best setting of each hyperparameter.

Models	Searching range of hyperparameters
LR	optimization algorithm: [**liblinear**, lbfgs, sag]
LASSO	*L*1 regularization term: [0.0001, **0.001**, 0.01, 0.1, 1.0]
SVM	regularization parameter: [0, 0.5, **1.0**, 1.5, 2.0], kernel type: [linear, poly, **rbf**], kernel coefficient: [**scale**, auto]
DT	split criterion: [**gini**, entropy], maximal depth: [10, 20, **30**, 40]
SdAE	learning rate: [0.0001, **0.001**, 0.01, 0.1], *L*2 regularization term: [0, 0.0001, **0.001**, 0.01, 0.1], number of denoising autoencoder layers: [2, **3**, 4], hidden dimension of each layer: [30, **40**, 50, 60]
TARPML	learning rate: [0.0001, **0.001**, 0.01, 0.1], *L*2 regularization term: [0, 0.0001, **0.001**, 0.01, 0.1], number of LSTM layers: [1, **2**, 3, 4], number of FC layers: [1, **2**, 3, 4], LSTM unit hidden dimension: [16, 32, **64**, 128], FC hidden dimension: [30, 40, **50**, 60]
DSTGCN	learning rate: [0.0001, **0.001**, 0.01, 0.1], *L*2 regularization term: [0, 0.0001, **0.001**, 0.01, 0.1], number of spatial convolutional blocks: [1, 2, **3**, 4], number of spatio-temporal convolutional blocks: [1, 2, **3**, 4]

5.2. Experimental results

We first compare our method with the baselines and then we evaluate the impact of different features. We also investigate the effect of the structure of our proposed model.

Model Comparison. To avoid occasional results, we run each model for 10 times repeatedly. The deep learning models (i.e., SdAE, TARPML and DSTGCN) are trained for 150 epochs in the training process and models which achieve the best performance on validation set are chosen for testing. We report the mean value $\pm$ standard deviation of metrics. The standard deviations of LR, LASSO, SVM vary in a very small range slightly and we denote them as 0. Experimental results are shown in Table 5:

From the results, some conclusions could be summarized.

Firstly, SVM shows better performance than LR and LASSO, because it designs the kernel trick to find the best line separator gap and is more powerful in learning complex non linear functions. Among the classical machine learning models, DT performs better than others in most metrics because DT is better at selecting more important features relevant to traffic accidents and less sensitive to noises in the inputs.

Secondly, TARPML performs better than SdAE because TARPML leverages the RNN structures to learn temporal information while SdAE ignores the dynamic impacts in temporal features.

Thirdly, the deep learning models obtain better performance that the classical machine learning models while demonstrates the ability of deep architectures in modeling complex relationships. And the standard deviations of deep learning models are varied in a small range, which show the stability of these models.

Finally, we could see that DSTGCN outperforms other methods on all the evaluation metrics. We contribute this phenomenon to two reasons: (1) DSTGCN designs suitable module to handle spatio-temporal information respectively, which provides DSTGCN

[3] https://pytorch.org.
[4] https://github.com/yule-BUAA/DSTGCN

Table 5
Comparisons with different methods.

Models		*RMSE*	*PCC*	*Precision*	*Recall*	*F1-Score*	*AUC*
Classical models	LR	0.4713 ± 0	0.5639 ± 0	0.7376 ± 0	0.8625 ± 0	0.7952 ± 0	0.7779 ± 0
	LASSO	0.4206 ± 0	0.5472 ± 0	0.7230 ± 0	0.8785 ± 0	0.7932 ± 0	0.7709 ± 0
	SVM	0.4541 ± 0	0.5985 ± 0	0.7471 ± 0	0.8884 ± 0	0.8116 ± 0	0.7938 ± 0
	DT	0.4120 ± 0.0075	0.6625 ± 0.0127	0.8059 ± 0.0059	0.8699 ± 0.0107	0.8367 ± 0.0063	0.8302 ± 0.0062
State-of- the-art	SdAE	0.3920 ± 0.0177	0.6644 ± 0.0229	0.7724 ± 0.0195	0.8890 ± 0.0246	0.8263 ± 0.0122	0.8130 ± 0.0139
	TARPML	0.3687 ± 0.0078	0.7127 ± 0.0120	0.8029 ± 0.0132	0.8942 ± 0.0134	0.8460 ± 0.0044	0.8372 ± 0.0059
Our model	DSTGCN	**0.3439 ± 0.0106**	**0.7445 ± 0.0169**	**0.8213 ± 0.0122**	**0.8968 ± 0.0166**	**0.8573 ± 0.0114**	**0.8508 ± 0.0117**

with the ability to consider not only spatial influences but also temporal correlations in heterogeneous data. (2) DSTGCN employs the embedding layer for removing noises and learning semantic representations of external information, which makes the model more robust. It is worth mentioning that the main difference of DSTGCN and existing methods is that DSTGCN takes the topological structure of roads into consideration, which helps the model utilize more comprehensive information and improve the final prediction performance.

Effects of Different Features. To investigate the effects of different features on model performance, we manually remove spatial, temporal and external features correspondingly and compute the evaluation metrics based on the remaining features. Table 6 shows the model performance when removing different features. From Table 6, we could see that removing any category of features would lead to the reduction in model performance. Spatial features reflect the building distributions and road structures, temporal features contain the dynamic changes in traffic flow and external features provide global attributes such as meteorological conditions and calendar representations. Taking all of the features as inputs would provide sufficient information for the proposed model to discover hidden influencing factors of traffic accidents and obtain better results.

Effects of Model Structures. To study the effects of model structures on model performance, we manually remove the spatial layer, spatio-temporal layer and embedding layer for external features respectively and conduct experiments on the remaining structures. The performance is shown in Table 7, where S, ST and E stand for the Spatial layer, Spatio-Temporal Layer and Embedding layer. From Table 7, we could conclude that each structure in DSTGCN has its own contribution to the model performance and removing any structures would achieve worse results. DSTGCN integrates the above structures together by employing spatial layers to describe the influence of spatial information, designing spatio-temporal layers to capture the dynamic changes in temporal information and leveraging the embedding layer to fuse external auxiliary information. All of the three structures work together and achieve the best performance.

To make the above experimental results more intuitive, Fig. 5(a) and (b) show the visualization of results as well.

5.3. Computation cost analysis

To compare the efficiency of different methods, we analyze the computation cost of each model in this subsection. The classical methods (LR, LASSO, SVM and DT) cost less time than the state-of-the-art methods (SdAE and TARPML) because they have much less parameters to train. However, they obtain unsatisfactory performance making them infeasible to be applied in predicting traffic accidents accurately and it is not necessary to estimate the computation costs of classical methods. Hence, we compare the computation cost of the proposed DSTGCN with SdAE and TARPML to show which model is more efficient. The experiments of estimating execution time are conducted on an Ubuntu machine equipped with two Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20 GHz with 8 physical cores, and the GPU is NVIDIA TITAN Xp, armed with 12 GB of GDDR5X memory running at over 11 Gbps. Experimental results are shown in Table 8.

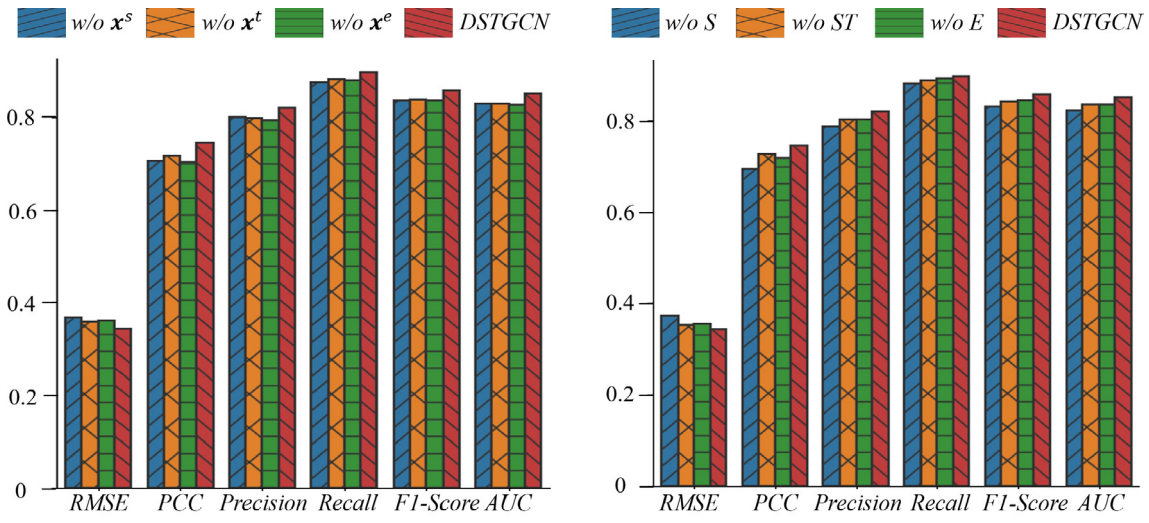
From the results, we could see that SdAE takes more time to train because it should first pre-train the stacked denoising autoencoders to learn hidden representations of the features and then leverages the labels to train its output layer. DSTGCN takes a bit more time than TARPML in each training epoch and they tend to converge in the same number of iterations roughly. Compared with TARPML, DSTGCN employs graph convolutions to consider the spatial correlations of a road and its neighbors, and the aggregation of spatial factors may lead to a little slower training speed of DSTGCN than that of TARPML. However, the proposed DSTGCN still have two advantages than TARPML. Firstly, which is more important, DSTGCN outperforms TARPML over a wide range of evaluation metrics. Secondly, the RNN structures in TARPML need to be processed sequentially, since the subsequent steps depend on the

Table 6
Effects of different features on prediction performance.

Feature	*RMSE*	*PCC*	*Precision*	*Recall*	*F1-Score*	*AUC*
$w/o\ \boldsymbol{x}^s$	0.3673	0.7050	0.7996	0.8765	0.8362	0.8283
$w/o\ \boldsymbol{x}^t$	0.3589	0.7162	0.7973	0.8829	0.8370	0.8283
$w/o\ \boldsymbol{x}^e$	0.3621	0.7030	0.7942	0.8809	0.8352	0.8261
$\boldsymbol{x}^{s+t+e}$	**0.3439**	**0.7445**	**0.8213**	**0.8968**	**0.8573**	**0.8508**

Table 7
Impacts of model structures on prediction performance.

Structure	*RMSE*	*PCC*	*Precision*	*Recall*	*F1-Score*	*AUC*
$w/o\ S$	0.3716	0.6948	0.7876	0.8825	0.8322	0.8221
$w/o\ ST$	0.3525	0.7280	0.8028	0.8884	0.8432	0.8349
$w/o\ E$	0.3550	0.7180	0.8019	0.8936	0.8452	0.8363
DSTGCN	**0.3439**	**0.7445**	**0.8213**	**0.8968**	**0.8573**	**0.8508**



(a) Effects of different features. (b) Impacts of model structures.

Fig. 5. Ablation experiments of the proposed model.

Table 8
Execution time and the number of converge epoch of different models.

Models	Training Cost Each Epoch (s)	Converge Epoch
SdAE	53	About 80
TARPML	21	About 60
DSTGCN	26	About 60

previous steps. However, DSTGCN does not use a recurrent architecture and the convolutions in DSTGCN could be faster by the parallel techniques (e.g., applying the same filter to multiple locations of the sequence at the same time).

6. Conclusion

In this paper, we studied the problem of traffic accidents and proposed a novel graph-based spatio-temporal model to predict the risk of future traffic accidents. To achieve this goal, a great amount of data, including traffic accident records, taxi GPS, POI distributions, meteorological observations as well as road networks were collected and relevant features were extracted. The proposed model consists of three parts: the spatial layer was designed to discover spatial correlations in spatial features; the spatio-temporal layer was utilized to capture both spatial relationships and temporal dependencies in temporal features; the embedding layer was leveraged to learn meaningful and dense representations of external features. Experiments on real-world datasets demonstrated the superiority of the proposed model over existing methods. Reducing the risk of traffic accidents is essential to urban transportation and public safety, and the proposed model could be applied to warn potential dangers in advance and help people choose safer traveling routes.

CRediT authorship contribution statement

Le Yu: Conceptualization, Methodology, Writing - original draft. **Bowen Du:** Supervision, Funding acquisition. **Xiao Hu:** Methodology, Writing - original draft. **Leilei Sun:** Conceptualization, Formal analysis. **Liangzhe Han:** Investigation, Visualization. **Weifeng Lv:** Funding acquisition, Funding acquisition, Resources.

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.

Acknowledgements

The authors would like to thank the reviewers for their constructive comments on this research work. This work is supported by the National Key R&D Program of China [Grant No. 2018YFB2101003], the Science and Technology Major Project of Beijing [Grant No. Z191100002519012], and the National Natural Science Foundation of China [Grant Nos. 51778033, 51822802, 51991395, 71901011, U1811463].

References

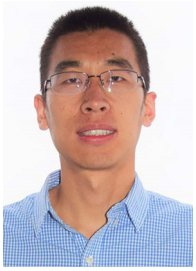
[1] Y. Chen, Y. Lv, Z. Li, F. Wang, Long short-term memory model for traffic congestion prediction with online open data, in: 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016, Rio de Janeiro, Brazil, November 1–4, 2016, 2016, pp. 132–137.
[2] Y. Kuboi, J. Imura, T. Hayakawa, H. Tanaka, Y. Mae, Traffic optimization via road pricing for queuing and flow congestion, in: 20th IEEE International Conference on Intelligent Transportation Systems, ITSC 2017, Yokohama, Japan, October 16–19, 2017, 2017, pp. 1–6.
[3] F. Costabile, I. Allegrini, A new approach to link transport emissions and air quality: An intelligent transport system based on the control of traffic air pollution, Environmental Modelling and Software 23 (2008) 258–267.
[4] X. Yu, W. Zhang, L. Zhang, V.O.K. Li, J. Yuan, I. You, Understanding urban dynamics based on pervasive sensing: An experimental study on traffic density and air pollution, Mathematical and Computer Modelling 58 (2013) 1328–1339.
[5] J.J. Rolison, S. Regev, S. Moutari, A. Feeney, What are the factors that contribute to road accidents? an assessment of law enforcement views, ordinary drivers' opinions, and road accident records, Accident Analysis & Prevention 115 (2018) 11–24.
[6] L. Gicquel, P. Ordonneau, E. Blot, C. Toillon, P. Ingrand, L. Romo, Description of various factors contributing to traffic accidents in youth and measures proposed to alleviate recurrence, Frontiers in Psychiatry 8 (2017) 94.
[7] L.-Y. Chang, W.-C. Chen, Data mining of tree-based models to analyze freeway accident frequency, Journal of Safety Research 36 (2005) 365–375.
[8] Y. Lv, S. Tang, H. Zhao, Real-time highway traffic accident prediction based on the k-nearest neighbor method, in: 2009 International Conference on Measuring Technology and Mechatronics Automation, vol. 3, 2009, pp. 547–550. doi: 10.1109/ICMTMA.2009.657.
[9] M. Hossain, Y. Muromachi, A bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways, Accident Analysis & Prevention 45 (2012) 373–381.

[10] L. Lin, Q. Wang, A. W. Sadek, A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction, Transportation Research Part C: Emerging Technologies 55 (2015) 444–459. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.
[11] Q. Chen, X. Song, H. Yamada, R. Shibasaki, Learning deep representation from big and heterogeneous data for traffic accident inference, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA., 2016, pp. 338–344.
[12] H. Ren, Y. Song, J. Liu, Y. Hu, J. Lei, A deep learning approach to the prediction of short-term traffic accident risk, CoRR abs/1710.09543 (2017).
[13] Z. Yuan, X. Zhou, T. Yang, Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19–23, 2018, 2018, pp. 984–992.
[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.
[15] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.
[16] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, 2015, pp. 2048–2057.
[17] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph neural networks: A review of methods and applications, arXiv preprint arXiv:1812.08434 (2018).
[18] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, arXiv preprint arXiv:1901.00596 (2019).
[19] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Processing Magazine 30 (2013) 83–98.
[20] A. Micheli, Neural network for graphs: A contextual constructive approach, IEEE Transactions on Neural Networks 20 (2009) 498–511.
[21] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden., 2018, pp. 3634–3640.
[22] C. Li, Z. Cui, W. Zheng, C. Xu, J. Yang, Spatio-temporal graph convolution for skeleton based action recognition, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[23] Y. LeCun, Y. Bengio, G.E. Hinton, Deep learning, Nature 521 (2015) 436–444.
[24] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA., 2017, pp. 1655–1661.
[25] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, X. Zhou, LC-RNN: A deep learning model for traffic speed prediction, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden., 2018, pp. 3470–3476.
[26] W. Cheng, Y. Shen, Y. Zhu, L. Huang, A neural attention model for urban air quality inference: Learning the weights of monitoring stations, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, 2018, pp. 2151–2158.
[27] X. Yi, J. Zhang, Z. Wang, T. Li, Y. Zheng, Deep distributed fusion network for air quality prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19–23, 2018, 2018, pp. 965–973.
[28] B. Wang, D. Zhang, D. Zhang, P.J. Brantingham, A.L. Bertozzi, Deep learning for real time crime forecasting, CoRR abs/1707.03340 (2017).
[29] A. Karduni, A. Kermanshah, S. Derrible, A protocol to convert spatial polyline data to network formats and applications to world urban road networks, Scientific Data 3 (2016) 1–7.
[30] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, 2017.
[31] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).
[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[33] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[34] S.H. Walker, D.B. Duncan, Estimation of the probability of an event as a function of several independent variables, Biometrika 54 (1967) 167–179.
[35] Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 267–288.
[36] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.
[37] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[38] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

Le Yu received the B.S. in computer science and engineering, Beihang University, China, in 2019. He is currently a first-year computer science Ph.D. student at Beihang University. His research interests include representation learning, graph neural networks and temporal data mining.

Bowen Du received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently a Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multi-source data fusion, and traffic data mining.

Xiao Hu received the B.S. in College of Software, Beihang University, China, in 2018. M.S. degree candidate in Computer Science and Engineering from Beihang University, Beijing, China. His research interests include intelligent transportation, deep learning and smart city technology.



Leilei Sun is an assistant professor of the State Key Laboratory of Software Development Environment and Big Data Brain Computing Lab (SKLSDE and BDBC Lab), Beihang University, Beijing, China. He was a postdoctoral research fellow from 2017 to 2019 in School of Economics and Management, Tsinghua University. He received his B.S. degree, in 2009, and M.S. degree, in 2012, from School of Control Theory and Control Engineering, Dalian University of Technology. He received his Ph.D. degree from Institute of Systems Engineering, Dalian University of Technology, in 2017. His research interests include machine learning and data mining. He has published many papers on IEEE Transactions on Data and Knowledge Engineering (TKDE), Knowledge and Information Systems (KAIS), and ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), etc.

Liangzhe Han received the B.S. in College of Software, Beihang University, China, in 2019. He is currently a Ph. D degree candidate in School of Computer Science and Engineering in Beihang University, Beijing, China. His research interests include intelligent transportation, deep learning and spatial temporal data mining.

Weifeng Lv received the B.S. degree in Computer Science and Engineering from Shandong University, Jinan, China, and the Ph.D. degree in Computer Science and Engineering from Beihang University, Beijing, China, in 1992 and 1998 respectively. Currently, he is a Professor with the State Key Laboratory of Software Development Environment, Beihang University, Beijing, China. His research interests include smart city technology and mass data processing.