Deep Graph Convolutional Networks for Incident-Driven Traffic Speed Prediction

Qinge Xie[1,2,3], Tiancheng Guo[1,2], Yang Chen[1,2,3], Yu Xiao[4], Xin Wang[1,2] and Ben Y. Zhao[5]
[1]School of Computer Science, Fudan University, China
[2]Shanghai Key Lab of Intelligent Information Processing, Fudan University, China
[3]Peng Cheng Laboratory, China
[4]Department of Communications and Networking, Aalto University, Finland
[5]Department of Computer Science, University of Chicago, USA
{qgxie17,tcguo16,chenyang,xinw}@fudan.edu.cn,
yu.xiao@aalto.fi,ravenben@cs.uchicago.edu

ABSTRACT

Accurate traffic speed prediction is an important and challenging topic for transportation planning. Previous studies on traffic speed prediction predominately used spatio-temporal and context features for prediction. However, they have not made good use of the impact of traffic incidents. In this work, we aim to make use of the information of incidents to achieve a better prediction of traffic speed. Our incident-driven prediction framework consists of three processes. First, we propose a critical incident discovery method to discover traffic incidents with high impact on traffic speed. Second, we design a binary classifier, which uses deep learning methods to extract the latent incident impact features. Combining above methods, we propose a Deep Incident-Aware Graph Convolutional Network (DIGC-Net) to effectively incorporate traffic incident, spatio-temporal, periodic and context features for traffic speed prediction. We conduct experiments using two real-world traffic datasets of San Francisco and New York City. The results demonstrate the superior performance of our model compared with the competing benchmarks.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; *Data mining*.

KEYWORDS

Real-time traffic prediction, deep neural network, time series, traffic incidents

ACM Reference Format:
Qinge Xie[1,2,3], Tiancheng Guo[1,2], Yang Chen[1,2,3], Yu Xiao[4], Xin Wang[1,2] and Ben Y. Zhao[5]. 2020. Deep Graph Convolutional Networks for Incident-Driven Traffic Speed Prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411873

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411873

1 INTRODUCTION

Traffic speed prediction has been a challenging problem for decades, which has a wide range of traffic planning and related applications, including congestion control [17], vehicle routing planning [14], urban road planning [28] and travel time estimation [9]. The difficulty of the prediction problem comes from the complicated and highly dynamic nature of traffic and road conditions, as well as a variety of other unpredictable, ad hoc factors. Traffic incidents, including lane restriction, road construction and traffic collision, which is one of the most important factors, tend to dramatically impact traffic for limited time periods. Yet the frequency of these events means their aggregate impact cannot be ignored when modeling and predicting traffic speed.

Despite a large amount of research on detecting traffic incidents [10, 40, 41], a small number of works have explored the impact of traffic incidents recently. Miller et al. [26] proposed a system for predicting the cost and impact of highway incidents. Javid et al. [13] developed a framework to estimate travel time variability caused by incidents. He et al. [11] proposed to use the ratio of speed before and after incidents as the traffic impact coefficient to evaluate the traffic influence of incidents. Those works have proven the significant impact of traffic incidents on traffic conditions. However, improving traffic speed prediction by traffic incidents has not been well explored. Some previous works [18, 19] used incident data collected from social networks (e.g., Twitter) by keywords to improve traffic prediction. However, they failed to consider the impact level of different traffic incidents but treat all incidents equally for speed prediction. The large majority solutions including traditional machine learning [4], matrix decomposition [7] and deep learning methods [16, 22, 37] of traffic speed prediction mainly used spatio-temporal features of traffic network and context features such as weather data. These solutions for predicting traffic speed do not factor in the impact of those dynamic traffic incidents.

A number of questions naturally arise: how do different traffic incidents impact traffic flow speeds? Do high impact traffic incidents show specific spatio-temporal patterns in a city? How can we use traffic incident data to improve traffic speed prediction? In this paper, our goal is to answer these questions, and to find an effective way to improve traffic speed prediction using traffic incident data. There are two main challenges for our incident-driven traffic speed prediction problem. First, the impact of traffic incidents is complex and varies significantly across incidents. For example, incidents

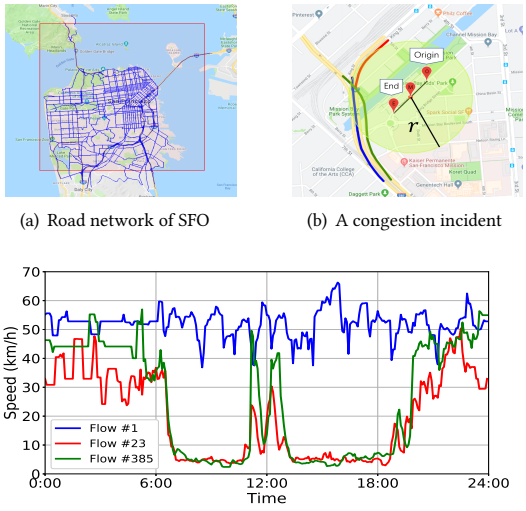
which occur early in the morning and in remote areas will have little impact on adjacent roads, while the ones which occur during the rush hours and in high-traffic areas (e.g. downtown) are very likely to affect the surrounding traffic flows or even cause congestion [27]. Therefore, it is unreasonable to treat all traffic incidents equally for traffic speed prediction, which may even negatively impact the prediction performance. Second, the impact of traffic incidents on adjacent roads will be affected by external factors like incident occurrence time, incident type and the road topology structure. We need to extract the latent impact features of traffic incidents to improve the traffic prediction.

To tackle the first challenge, we propose a critical incident discovery method to quantify the impact of urban traffic incidents on traffic flows. We consider both anomalous degree and speed variation of adjacent roads to discover the critical traffic incidents. Next, to tackle the second challenge, we propose a binary classifier which uses deep learning methods to extract the latent impact features of incidents. The impact of incidents varies in degree and the impact is neither binary nor strict multi-class. So we extract the latent impact features from the middle layer of the classifier, where the latent features are continuous and filtered. We adopt Graph Convolution Networks (GCN) [3] to capture spatial features of road networks. GCN is known to be able to effectively capture the topology features in non-Euclidean structures and the complex road network is a typical non-Euclidean structure. Combining aforementioned methods, we propose a Deep Incident-Aware Graph Convolutional Network (DIGC-Net) to improve traffic prediction by utilizing traffic incident data. DIGC-Net can effectively leverage traffic incident, spatio-temporal, periodic and context features of a city for prediction.

We test our framework using two real-world urban traffic datasets of San Francisco and New York City. Experimental results empirically answer the aforementioned questions, and also show the particularly different spatio-temporal distributions of critical/non-critical incidents. We compare DIGC-Net with the state-of-the-art methods [5–7, 12, 22, 30, 31], and the results demonstrate the superior performance of our model and also verify that the incident learning component is the key to the improvement of prediction performance.

We summarize our key contributions as follows:

- To quantify the impact of traffic incidents on traffic speeds, we propose a critical incident discovery method to discover critical incidents in a city. We further explore the spatio-temporal distributions of critical/non-critical incidents and find noteworthy differences.
- In order to extract the latent incident impact features, we design a binary classifier to extract the latent impact features from the middle layer of the classifier. We use the binary classifier as an internal component of our final framework to improve traffic speed prediction.
- We propose DIGC-Net to effectively incorporate incident, spatio-temporal, periodic and context features of a city for traffic speed prediction. We test our framework using two real-world urban traffic datasets, and the incident learning component of our framework can be flexibly inserted into other models for learning incident impact features.



(a) Road network of SFO (b) A congestion incident

(c) Speed curves of three candidate flows

Figure 1: Traffic illustration of SFO

2 RELATED WORK

Traffic Speed Prediction. A number of solutions have been proposed for traffic speed prediction. ARIMA [5] is a classical model for this area, and regression methods [4] are also widely used for predicting traffic speed. There are also matrix spectral decomposition models for traffic speed prediction: [7] proposed a latent space model to capture both topological and temporal properties. Recently, deep learning approachs achieve great success in this space by using spatio-temporal and context features [21, 24]. The spatio-temporal and context structure is a common use in traffic prediction. Zhang et al. [42] divided road network into grids and used CNN to capture spatial dependencies. Lv et al. [22] proposed a model that integrates both RNN and CNN models. GCN begins to be used for traffic speed prediction recently because of the ability to effectively capture the topology features in non-Euclidean structures. Li et al. [16] proposed to model the traffic flow as a diffusion process on a directed graph. Yu et al. [38] proposed the STGCN model to tackle the time series prediction problem in traffic domain. Zheng et al. [44] proposed a graph multi-attention network (GMAN) to predict long-term traffic conditions. In our work, we effectively incorporate traffic incident, spatio-temporal, periodic and weather features of a city for traffic speed prediction. Our main contributions focus on the effective utilization of incident information for improving prediction performance.

Urban Incidents. Research on urban anomalous incidents mainly focus on the detection of incidents. Gu et al. [10] mined tweet texts to extract incident information to do the traffic incident detection. Zhang et al. [41] proposed an algorithm based on SVM to capture rare patterns to detect urban anomalies. Yuan et al. [40] proposed a ConvLSTM model for traffic incident prediction. There are also a few works focus on mining the impact of incidents. Miller et al. [26] proposed a system for predicting the cost and impact of highway incidents, in order to classify the duration of the incident induced

delays and the magnitude of the incident impact. Javid et al. [13] developed a framework to estimate travel time variability caused by traffic incidents by using a series of robust regression methods. In our work, we extract the latent incident impact features for traffic speed prediction.

3 PRELIMINARIES

Before diving into details of DIGC-Net, we begin with some preliminaries on our datasets and problem formulation.

3.1 Datasets

We utilize two datasets, a traffic dataset and an attribute dataset (weather data). The traffic dataset consists of traffic road network, speed and incident sub-dataset of two major metropolitan areas, San Francisco (SFO) and New York City (NYC), with complex traffic conditions and varying physical features that may affect latent traffic patterns [34]. In Section 4 and Section 5, we use the traffic incident, road network and speed sub-datasets. The incident and speed data covers the time range of Apr. 17 to Apr. 24, 2019. In Section 6, we use traffic incident, road network, speed sub-datasets and the weather dataset. The incident and speed data were collected from Apr. 4 to May 2, 2019 (4 weeks). We collected the weather dataset by Yahoo Weather API [35] and the fields includes weather type, temperature and sunrise time. We collected the traffic dataset from a public API: HERE Traffic [1]. The dataset consists of: 1) Road Network: We set latitude/longitude bounding boxes (Figure 1(a)) on two cities of SFO (37.707,-122.518/37.851,-122.337) and NYC (40.927,-74.258/40.495,-73.750) to gather the internal road networks. 2) Traffic Speed: We collected the real-time traffic speed of each flow in the areas described above and record real-time speeds of each flow every 5 minutes. 3) Traffic Incident: We also collected the traffic incident data in same areas every 5 minutes. For each incident, we could get the incident features like type and location.

Flow. Real-time speeds in different segments of single road are discrete. HERE divides every road into multiple segments. We denote one road segment as one flow ξ. Every flow at each time slot has a speed and we use flow as the smallest unit of the road network.

3.2 Problem Formulation and Preprocessing

First, we denote a road network as an undirected graph $\mathcal{N} = (V, E)$, where each node represents an intersection or a split point on the road, and each edge represents a road segment.

Reconstruction of the road network. As our task is to predict the speed of every road segment, we use the road segment as the node. More specifically, we use every flow as one node to build the road network. If two flows ξ_i and ξ_j have points of intersection, we will add an edge to connect node ξ_i and node ξ_j. Therefore, we build a new road network graph $\mathcal{G} = (V, E)$, where each node represents a flow and each edge represents an intersection of the flows or a split point on the flow. There are 2,416 nodes and 19,334 edges of the SFO graph, and 13,028 nodes and 92,470 edges of the NYC graph. We will use the re-build road network graph $\mathcal{G}$ in the rest of the paper.

Problem formulation. We use $v^t_{\xi_i}$ to represent the speed of flow ξ_i at time slot t. For every speed snapshot of the road network, we will get a vector of all flows $V^t = \left[v^t_{\xi_0}, v^t_{\xi_1}, \cdots, v^t_{\xi_{N-1}}\right]$, where N is the total number of flows. Given the re-build road graph $\mathcal{G} = (V, E)$ and a T-length historical real-time speed sequence $[V^{t-T}, V^{t-T+1}, \cdots, V^{t-1}]$ of all flows, our task is to predict future speeds of every flow in the city, *i.e.*, $Y = \left[V^t, V^{t+1}, \cdots, V^{t+k-1}\right]$, where k is the prediction length. Given a set of urban traffic incidents occur close to the predicted time t, more specifically, a set of incidents occur within $[t - T_1, t - T_2]$, where $t - T_1$ is the earliest included incident occurrence time and $t - T_2$ is the latest included incident occurrence time. We extract the features of the impact of aforementioned incidents on traffic flows to improve the speed prediction performance.

4 URBAN CRITICAL INCIDENT DISCOVERY

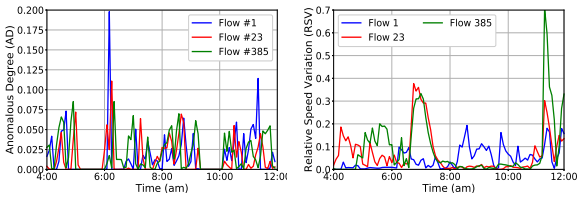
The impact of urban traffic incidents are complex and also influenced by other factors like the topological structure of urban road network, temporal features of traffic conditions and the incident type. Treating all urban traffic incidents equally will add additional noise to traffic speed prediction process. In this section, we focus on analyzing the impact of different urban traffic incidents, and introduce our urban critical incident discovery methodology.

4.1 Methodology

Case Study: A Congestion Incident. Figure 1(b) presents a congestion incident occurred at 06:32 am on Apr. 17, 2019 in San Francisco. M is the center point of the incident and we set r to represent the radius of the impact range. The circle with the center M and radius r stands for the region affected by the incident. We define that if the center of flows is in the circle, then the flows might be affected by the incident. The circle in Figure 1(b) presents the affected region when $r = 300\,m$. The blue, red and green lines represent three flows ξ_1, ξ_{23} and ξ_{385} in San Francisco which might be affected by the incident, respectively. The speed curves of the three candidate flows are shown in Figure 1(c). We observe that during 6:00 am - 7:00 am, the speeds of ξ_{23} and ξ_{385} show a sharp reduction while the variation of ξ_1 is relatively slight, but it still become more choppy after the incident occurred.

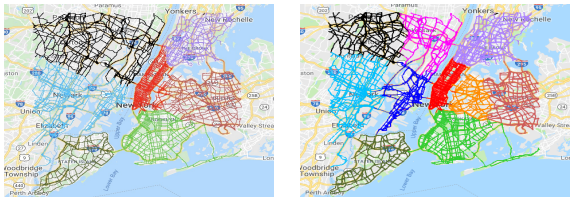
Next, we analyze each candidate flow that whether it will truly be affected by the incident. We use a variant of the method proposed in [41] to compute the anomalous degree of each flow. They divided the city area into several grids and computed the anomalous degree of each grid region to detect urban anomalies. The key idea to compute the anomalous degree of a region is based on its historically similar regions in the city. The sudden drop of speed similarity of a region and its historically similar regions indicates the occurrence of urban anomalies, and the well-designed experiments in [41] had verified the effectiveness of the detection method. In our problem, we use each flow as the unit rather than grid region.

Definition 1. Pair-wise Similarity of Flows. Given two flows at time slot t with speeds $v^t_{\xi_i}$ and $v^t_{\xi_j}$, for a time window $W = [t - T + 1 : t]$, the pair-wise similarity is calculated by: $s^{[t-T+1:t]}_{\xi_i,\xi_j} =$



(a) Anomalous degree $\mathcal{A}$ (b) Relative speed variation $\mathcal{R}$

Figure 2: Anomalous degree and relative speed variation of three candidate flows



(a) $k = 7$ (b) k=10

Figure 3: Clusters of NYC

$P\left(v_{\xi_i}^{[t-T+1:t]}, v_{\xi_j}^{[t-T+1:t]}\right)$, where P is to calculate Pearson correlation coefficient [20] of two speed sequences. Then the similarity matrix $\mathcal{S}$ of all flows at t is calculated by the following equation:

$$\mathcal{S}^t = \begin{vmatrix} s_{\xi_0,\xi_0}^{[t-T+1:t]} & \dots & s_{\xi_0,\xi_N-1}^{[t-T+1:t]} \\ \dots & \ddots & \dots \\ s_{\xi_N-1,\xi_0}^{[t-T+1:t]} & \dots & s_{\xi_N-1,\xi_N-1}^{[t-T+1:t]} \end{vmatrix}, \tag{1}$$

where N is the total number of flows in the city.

Definition 2. Similarity Decrease Matrix $\mathcal{D}$. Similar to [41], we define the similarity decrease matrix $\mathcal{D}$, which represents the decreased similarity of each flow pair from time slot $t-1$ to t. $\mathcal{D}$ at time slot t is calculated by: $\mathcal{D}^t = max\left(0, S^{t-1} - S^t\right)$. Zeroing the numbers less than zero is due to that we only consider the case where the similarity goes down.

Definition 3. Anomalous Degree $\mathcal{A}$. Then we use similarity matrix $\mathcal{S}$ and similarity decrease matrix $\mathcal{D}$ to compute anomalous degree of flows at time slot t. We use a threshold parameter δ to capture the historically similar flows. When the similarity of two flows is greater than or equal to δ, we define that they are historically similar. Given a flow ξ_i at time slot t, the historically similar flow sets of ξ_i is denoted as $\mathcal{H}_{\xi_i}^t = \{\xi_j \mid i \neq j \; and \; \mathcal{S}_{i,j}^t = \mathcal{S}_{j,i}^t \geq \delta\}$. Pair-wise similarity is computed by Pearson correlation coefficient (PCC) and PCC in [0.5, 0.7] indicates variables are moderately correlated according to [29]. Therefore, we set $\delta = 0.5$ here to select the historically similar flows which are at least moderately similarity to the flow ξ_i. Anomalous degree of flow ξ_i at time slot t is calculated by the following equation:

$$\mathcal{A}_{\xi_i}^t = \frac{\Sigma_{\xi_j \in \mathcal{H}_{\xi_i}^t} \mathcal{S}_{i,j}^{t-1} \cdot \mathcal{D}_{i,j}^t}{\Sigma_{\xi_j \in \mathcal{H}_{\xi_i}^t} \mathcal{S}_{i,j}^{t-1}}, \tag{2}$$

where $\mathcal{A}$ is the decrease degree in speed similarity of ξ_i and its historically similar flows.

Local Anomalous Degree Algorithm. The time complexity of computing similarity matrix $\mathcal{S}$ is $O\left(N^2\dot{T}\right)$, where N is the number of flows and T is the length of historical speed sequences. For cities with complex traffic road networks such as New York City (13,028 flows), it will cost a lot to compute the similarity matrix $\mathcal{S}$, the similarity decrease matrix $\mathcal{D}$ and the anomalous degree $\mathcal{A}$. We propose a local anomalous degree algorithm to speed up our method based on the spectral clustering algorithm [39]. Spectral clustering is able to identify spatial communities of nodes in graph structures. According to several studies [32, 36, 45], which assume that traffic in nearby locations should be similar, we also assume that flows in the same community and in the spatially nearby regions will be historically similar. Given a graph $\mathcal{G}$, we perform spectral decomposition and obtain k graph spatial features of each flow. Then we use K-means [8], a common unsupervised clustering method, to cluster flows into k classes.

ALGORITHM 1: Local Anomalous Degree Algorithm

Input: Road graph $\mathcal{G}$

1. Compute the adjacency matrix A, degree matrix D, and normalized Laplacian matrix $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.
2. Compute the first k eigenvectors $v_1, v_2, ..., v_C$ of L.
3. Let $F \in \mathbb{R}^{N \times k}$ is the feature matrix of all nodes in the graph.

for *node i in G* **do**

$F_i = \left[v_{0,i}, v_{1,i}, ..., v_{k-1,i}\right]$

end

4. Use K-means method to cluster nodes into k classes (k labels).
5. Compute local-similarity matrix $\mathcal{S}$ and local-similarity decrease matrix $\mathcal{D}$.

$$S_{\xi_i,\xi_j}^{[t-T+1:t]} = \begin{cases} 0 & , label_i \neq label_j \\ P\left(v_{\xi_i}^{[t-T+1:t]}, v_{\xi_j}^{[t-T+1:t]}\right) & , label_i = label_j \end{cases}$$

6. Compute local-anomalous degree $\mathcal{A}$.

$$\mathcal{A}_{\xi_i}^t = \frac{\Sigma_{\xi_j \in \mathcal{H}_{\xi_i}^t \& (label_j = label_i)} \mathcal{S}_{i,j}^{t-1} \cdot \mathcal{D}_{i,j}^t}{\Sigma_{\xi_j \in \mathcal{H}_{\xi_i}^t \& (label_j = label_i)} \mathcal{S}_{i,j}^{t-1}}$$

Validation of Local Algorithm. Figure 3 shows the clustering result when $k = 10$ and $k = 7$ (marked by different colors). The result shows that the eigenvectors can effectively capture spatial graph features. Figure 3(b) shows that our method divides New York City into 10 local districts which are conform to the real-world urban districts, e.g., the red area corresponds to the Manhattan area in New York City. Meanwhile, the results of choosing different k are similar and we set $k = 10$ here. Then we only need to compute the local values of the similarity matrix $\mathcal{S}$, the similarity decrease matrix $\mathcal{D}$ and the anomalous degree $\mathcal{A}$ in the same district.

Next, different from anomaly detection, we aim at exploring the impact on traffic flows of different urban traffic incidents. Also taking Figure 1(b) as an example, there is a flawed scene that three

flows ξ_1, ξ_{23} and ξ_{385} are historically similar to each other at time slot t. Therefore, the sharp variations of ξ_{23} and ξ_{385} will strongly affect the anomalous degree of ξ_1. Figure 2(a) shows the anomalous degrees of them from 4:00 am to 12:00 pm. Near 06:32 am, ξ_1 actually has a higher anomalous degree (0.198) than ξ_{23} (0.110) and ξ_{385} (0.085). However, we can see it intuitively in Figure 1(c) that when close to 06:32 am, the anomalous variation of speeds of ξ_{23} and ξ_{385} are more striking than ξ_1. The reason for this diametrically opposite result is that after the incident, the tendency of anomalous changes of ξ_{23} and ξ_{385} are mighty similar, which leads to the low anomalous degree of them. Therefore, in order to handle the aforementioned scenario, we add another metric to help amend our discovery method.

Definition 4. Relative Speed Variation $\mathcal{R}$. Given a flow ξ_i at time t, and the historical speed sequence $\left[v_{\xi_i}^{t-T+1}, v_{\xi_i}^{t-T+2}, \cdots, v_{\xi_i}^{t}\right]$ of ξ_i in a T-length time window, we define the relative speed variation of ξ_i as follow:

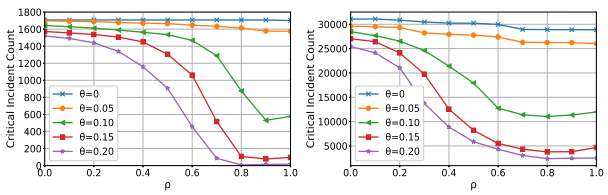
$$\mathcal{R}_{\xi_i}^t = \left| \frac{\sum_{t'=t-T+1}^{t'=t} v_{\xi_i}^{t'}}{T} - v_{\xi_i}^t \right| / max\left(v_{\xi_i}^{t_s}, v_{\xi_i}^{t_s+1}, \cdots, v_{\xi_i}^{t_e}\right) \quad (3)$$

We define a normalization time window and use the max value observed in the time window to normalize $\mathcal{R}$. We use 24 hours (288 intervals) as the normalization window length, i.e., $t_s = t - 144$ and $t_e = t + 144$, and $T = 10$ intervals.

Validation of Relative Speed Variation. As a heuristic approach, we test different candidate computing methods of relative speed variation as baselines for validation. We consider three related features: slope of speed variation (k) [33], recent speed (v^{t-1}) and historical average speed ($\bar{v}$) [2] corresponding to three candidate computing methods of Relative Speed Variation $\mathcal{R}$. They are listed as follows:

1) Consider all three features: $\mathcal{R}_{k+v^{t-1}+\bar{v}} = \left|\bar{v} - v^t\right| \times \bar{k} \times p + \left|v^{t-1} - v^t\right| \times k^{t-1} \times q$, where p and q are two parameters to control the ratio of recent speed and historical average speed. $\bar{k}$ is the historical average slope and k^{t-1} is the speed slope of time slot $t-1$ and t.
2) Consider recent speed and historical average speed: $\mathcal{R}_{v^{t-1}+\bar{v}} = \left|\bar{v} - v^t\right| \times p + \left|v^{t-1} - v^t\right| \times q$.
3) Consider historical average speed: $\mathcal{R}_{\bar{v}} = \left|\bar{v} - v^t\right|$.

We use the normalized item to normalize the three computing methods. We use Pearson correlation coefficient to calculate the correlation coefficient of anomalous degree and relative speed variation of all urban traffic incidents in our dataset (an hour before and after the incident). In order to use relative speed variation to amend anomalous degree, we choose the most negatively correlated computing method as our relative speed variation (p and q are set to 0.5), i.e., only consider the historical average speed: $\mathcal{R}_{\bar{v}} = \left|\bar{v} - v^t\right|$. Figure 2(b) shows the result of the congestion incident. Near 06:32 am, in contrast to $\mathcal{A}$, the max $\mathcal{R}$ of ξ_{23} and ξ_{385} are both larger (0.377 and 0.333) than ξ_1. It is conform to the speed variation (Figure 1(c)) and indicates that relative speed variation can also capture anomalies well and effectively correct the flaw of anomalous degree.



(a) SFO (b) NYC

Figure 4: Varying ρ and θ

Definition 5. Incident Effect Score $\mathcal{E}$. Due to the complementarity of anomalous degree and relative speed variation, we combine both of them to compute the incident effect score. Given a flow ξ_i at time slot t, the incident effect score is calculated by:

$$\mathcal{E}_{\xi_i}^t = \rho \cdot \mathcal{A}_{\xi_i}^t + (1-\rho) \cdot \mathcal{R}_{\xi_i}^t, \quad (4)$$

where ρ is a parameter to control the ratio of $\mathcal{A}$ and $\mathcal{R}$.

Definition 6. Critical Incidents. For incidents like mega-events, the traffic flows might be affected before incidents begin. On the contrary, incidents like traffic collisions will begin to affect traffic flows after they occurred. Therefore, given an incident $inci_k$ with a start time t_s, we firstly set a T-length "start to influence" window $W = \left[t_s - \frac{T}{2}, t_s + \frac{T}{2}\right]$ and define the flows which are highly affected by the incident as $\{\xi_i \mid max(\mathcal{E}_{\xi_i}^{t-\frac{T}{2}}, \mathcal{E}_{\xi_i}^{t-\frac{T}{2}+1}, \cdots, \mathcal{E}_{\xi_i}^{t+\frac{T}{2}}) \geq \theta\}$, where θ is a threshold parameter.

When $\left|\{\xi_i \mid max(\mathcal{E}_{\xi_i}^{t-\frac{T}{2}}, \mathcal{E}_{\xi_i}^{t-\frac{T}{2}+1}, \cdots, \mathcal{E}_{\xi_i}^{t+\frac{T}{2}}) \geq \theta\}\right|_{\mathcal{I}_k} > 0$, more specifically, there is at least one flow is highly affected by the incident $\mathcal{I}_k$, we call $\mathcal{I}_k$ is a critical incident, where $|\cdot|$ denotes the cardinality of a set. We define an incident which is not a critical incident as a non-critical incident.

4.2 Evaluation and results

Parameter Setting. The datasets we use here are listed in Section 3. We set $r = 500\,m$ and one hour as the length of "start to influence" time window.

Varing ρ and θ. Figure 4 shows the number of critical incidents discovered when varying ρ and θ. In SFO, when $\theta = 0$, most incidents are discovered as critical (1,706 out of 1,832 averagely), which indicates that most incidents indeed have an impact on traffic flows. There are a small number of incidents which almost have no impact (6.9%, $\theta = 0$ and 12.2%, $\theta = 0.05$), which further proves that treating all traffic incidents equally for traffic speed prediction is unreasonable. When θ rises ($\theta = 0.10$, 0.15 or 0.20), there is a sharp reduction of critical incidents, which indicates the impact of incidents varies in degree. In order to discover incidents with high impact, we set $\rho = 0.6$ and $\theta = 0.15$ of SFO. The results of NYC are similar with SFO. Most incidents are discovered as critical incident when θ is set to 0 or 0.05. Reductions also appear when θ rises. We set $\rho = 0.5$ and $\theta = 0.10$ of NYC.

Spatio-temporal Distributions. Figure 5 shows the spatial and temporal distributions of incidents in SFO and NYC. In Figure 5(a)

and Figure 5(c), an incident is plotted as a line with an origin and an end, and we find that although most of incidents of these two types occur on the main roads (continuous parts), our method can effectively discover critical incidents (green circle). Figure 5(b) and 5(d) show the temporal distributions. Incidents mostly occur within rush hours (7-9 am and 4-7 pm) of SFO and NYC, which is in line with the daily routine. Incidents which occur in the early morning tend to be non-critical in both cities. On the weekend, NYC only has one peak of incident occurrence (mid-afternoon) and on the weekday, NYC does not have the mid-afternoon peak while SFO presents the peak.

Summary of Results. Parameters ρ and θ represent the threshold to discover urban incidents with high impact on traffic speeds. The lower the θ and ρ are, the lower the threshold to mark critical incidents. The results of varying ρ and θ show that some urban incidents almost have no impact on traffic speeds and the impact of urban incidents varies in degree, which indicate that it is unreasonable to use all urban traffic incidents features for traffic speed prediction. Spatio-temporal distributions show noteworthy differences between urban critical and non-critical incidents, which indicates that our critical incident discovery method can effectively discover incidents with high impact on traffic speeds.

5 EXTRACT THE LATENT INCIDENT IMPACT FEATURES

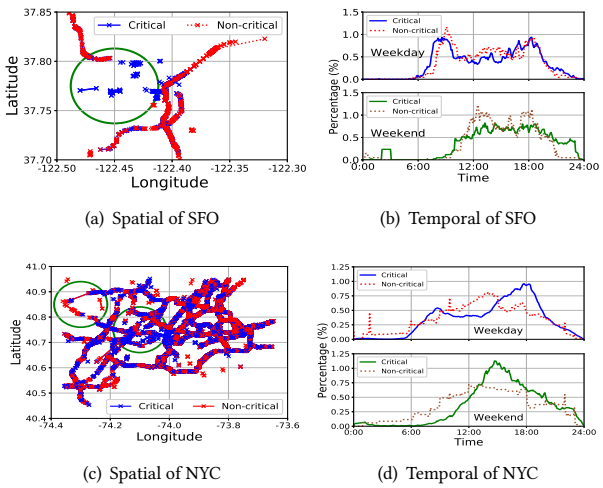
So far, we have demonstrated that our discovery method can effectively discover urban critical/non-critical incidents. In this section, we propose to use deep learning methods to extract the latent incident impact features for traffic speed prediction. Taking two aspects into account, we design a binary classifier to make use of the latent impact features:

- There are some urban incidents have almost no impact on traffic flows and low-impact incidents features will even bring noise to the model. There are also noteworthy differences of spatio-temporal features between crucial and non-crucial incidents, which inspires us to consider the binary classification problem.
- The impact of urban incidents on traffic speeds varies in degree and the impact is neither binary nor strict multi-class. Therefore, we should not use the binary result directly, we propose to extract the latent impact features from the middle layer of the binary classifier for traffic speed prediction, where the latent features are continuous and filtered.

5.1 Methodology

The task of the binary classifier is to predict whether an incident is critical/non-critical, i.e., whether an incident has a high/low impact on traffic speed. Considering that the impact of incidents is related to spatio-temporal and context features, and previous works [22, 36, 42] which use spatio-temporal and context features for traffic prediction (we also discuss them in Section 2), our classifier consists of three components: spatial learning component (GCN), temporal learning component (LSTM) and context learning component.

Spatial Learning: GCN (Figure 6(a)). City road networks has latent traffic patterns and there are complex spatial dependencies [16].



(a) Spatial of SFO (b) Temporal of SFO

(c) Spatial of NYC (d) Temporal of NYC

Figure 5: Spatio-temporal distributions of traffic incidents

We need to capture the road topological features, i.e., the spatial dependencies of the road network. Traditional methods divide the city into several grids and use Convolutional Neural Network (CNN) to capture spatial features [36, 42]. However, it neglects the road topological features and also ignores the spatial information within grids. Moreover, graph structure related features are hard to be used in CNN of our problem. We adopt graph convolutional network (GCN) [3] to learn the spatial topology features. GCN is known for being able to capture the topology features in non-Euclidean structures, which is suitable for road networks. GCN model $f(X, A)$ follows the layer-wise propagation rule [15]:

$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \tag{5}$$

where A is the adjacency matrix, $\tilde{A}$ is the adjacency matrix of the graph with added self-connections, D is the degree matrix and $\tilde{D}_{ii} = \Sigma_j \tilde{A}_{ij}$. $L = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the normalized Laplacian matrix of the graph $\mathcal{G}$. σ denotes an activation function. W is the trainable weight matrix, $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the l-th layer. $H^{(0)} = X$, where X is the input vectors of GCN.

We use the aforementioned graph $\mathcal{G}$. At each time slot t, we obtain a real-time speed of every flow in $\mathcal{G}$, and we define the speed snapshot $G^t = \{V^t_{\xi_0}, V^t_{\xi_1}, ..., V^t_{\xi_{N-1}}\}$, where N is the total number of flows in the city. We also add another graph structure related feature: the distance of each flow from the incident, which is because that the impact of incidents on flows has a strong correlation with distance [32, 36, 45]. We define the distance D_{ξ_i} of ξ_i from the incident is the Euclidean distance between the flow center and incident center. Therefore, at each time slot t, the input features $X = \left[\left(V^t_{\xi_0}, D_{\xi_0}\right), \left(V^t_{\xi_1}, D_{\xi_1}\right), ..., \left(V^t_{\xi_{N-1}}, D_{\xi_{N-1}}\right)\right]$. For a traffic incident, the time span of input speed snapshots is $\left[t_s - \frac{T}{2}, t_s + \frac{T}{2}\right]$, where t_s is the start time of the incident and T is the length of "start to influence" time window which is defined in Section 4.

For the input signal $X \in \mathbb{R}^{N\times C}$ with C input channels ($C = 2$ here) and F filters or features of spectral convolutions map are as

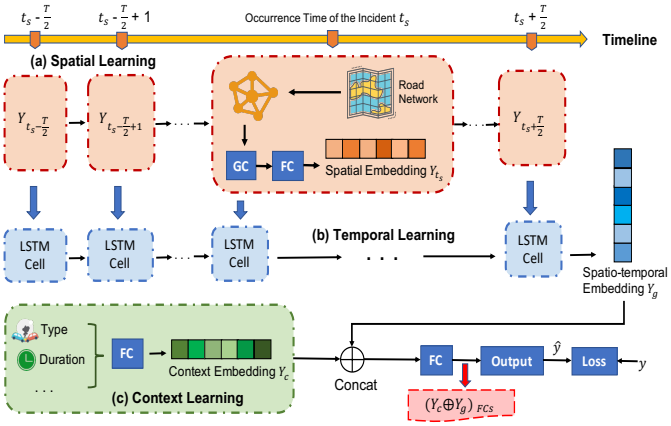


Figure 6: The architecture of the binary classifier

follow:

$$Z = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta, \tag{6}$$

where $\Theta \in \mathbb{R}^{C\times F}$ is a matrix of filter parameters, $Z \in \mathbb{R}^{N\times F}$ is the convolved signal matrix and F is the number of filters or features. Next, at each time slot t, after k graph convolutional (GC) layers, we then feed middle states H_t^k into m fully connected (FC) layers to get the spatial learning output Y_t of each snapshot.

Temporal Learning: LSTM (Figure 6(b)). We feed a sequence of graph speed snapshots to GCN, and the output is a sequence of spatial features at each time slot from $t_s - \frac{T}{2}$ to $t_s + \frac{T}{2}$. Then we adopt Long Short-Term Memory (LSTM) model [12] as our temporal learning component. LSTM is known for being able to learn long-term dependency information of time related sequences. LSTM has the ability to remove or add information to the state of the cell through a well-designed structure "gate". we extract the spatial features Y_t for each snapshot in GCN and feed the sequence $\left[Y_{t_s-\frac{T}{2}}, Y_{t_s-\frac{T}{2}+1}, \cdots, Y_{t_s+\frac{T}{2}}\right]$ into LSTM cells. Then we can iteratively get the output sequence $\left[h_{t_s-\frac{T}{2}}, h_{t_s-\frac{T}{2}+1}, \cdots, h_{t_s+\frac{T}{2}}\right]$. We use the output of the last LSTM cell output as the output Y_g of temporal learning part.

Context Learning (Figure 6(c)). Incident context features are also important for prediction. We use the following features for context learning: 1) Incident type (e.g., traffic collision and event). 2) Road status: An incident leads to a road close or not. 3) Start and end hour: HERE gives a start time t_s and an anticipative end time t_e of an incident. 4) Incident duration: The anticipative duration of an incident. 5) Weekday, Saturday or Sunday. We use one-hot encoding to preprocess class features and normalize the incident duration feature. The context learning component is a Deep Neural Network (DNN) structure, more specifically, an input layer and a fully connected layer (shown in Figure 6(c)). After embedding the context information, we feed the context embedding to a fully connected layer to get Y_c, which is the output of context learning.

Latent incident impact features extraction. After getting Y_c and spatio-temporal feature Y_g, we use a concat operation to concatenate them as $Y_c \odot Y_g$ of each incident. Then we feed $Y_c \odot Y_g$ to m FC layers. We extract the output of the last FC layer before the output layer as the latent incident impact features, which is because that output layer uses these features as the input to predict whether the incident has high impact on traffic flows. We denote the latent impact features as $(Y_c \oplus Y_g)_{FCs}$. Finally we get the prediction value $\hat{y}$, and compute the loss compared with real value y.

Objective Function and Evaluation Metric. The classifier is training by minimizing Binary Cross Entropy Loss (BCELoss) between the predicted speed and the real value. BCELoss is defined as $BCELoss = -(y \cdot log(\hat{y}) + (1-y) \cdot log(1-\hat{y}))$. We use BCELoss and $F1 - score = \frac{2 \cdot precision \cdot recall}{precision + recall}$ to evaluate the binary classifier.

5.2 Middle Experiments

Parameter Setting. The datasets we use here are described in Section 3. We use the discovery results obtained in the last section as the ground truth. There are 1,061 positive samples (critical) and 771 negative samples (non-critical) of SFO and 17,924 positive samples and 15,367 negative samples of NYC. We use 5 minutes as the time interval and train our classifier with the following hyper-parameter settings: learning rate (0.001) with the Adam optimizer. In GCN, we set two GCN layers followed by one FC layer with the 64-dimension output. The length of "start to influence" window is set to one hour, i.e., the input size of the first GCN layer is 12. We use *ReLU* activation function and add Dropout ($d = 0.8$) in GC layer. We use one LSTM layer with 64-dimension hidden states. After concating, we adopt one FC layer (16-dimension) and follow by the output FC layer using *sigmoid* activation function. We use 70% data for training and validation, and the remaining 30% as the test set. We select 90% of training set for training and 10% as the validation set for early stopping.

Results and Analysis. Using the traffic incident and traffic speed sub-datasets for training, we finally get 0.8241 F1-score and 0.4429 BCELoss in the test set of SFO, and 0.4731 BCELoss and 0.8000 F1-score of NYC. Our binary classifier model can capture the latent impact features on traffic flows of different incidents, more specifically, we can get the embedding $Y_c \oplus Y_g$ of each input incident. Y_c is the output features of context learning and Y_g is the output features of spatio-temporal learning. And we feed $Y_c \oplus Y_g$ into m ($m = 1$ in our experiment) FC layers to extract the latent impact features $(Y_c \oplus Y_g)_{FCs}$ before the ouput layer. We will use the binary classifier in the next section as an internal component to help improve traffic speed prediction performance. Since we take the classifier as a middleware of our incident-driven framework, we further evaluate our complete framework with competitive baselines in the next section.

6 INCIDENT-DRIVEN TRAFFIC SPEED PREDICTION

So far, we can effectively capture the latent impact features of urban incidents on traffic flow speeds. Combining above methods, we propose Deep Incident-Aware Graph Convolutional Network (DIGC-Net) to improve traffic speed prediction by incident data.

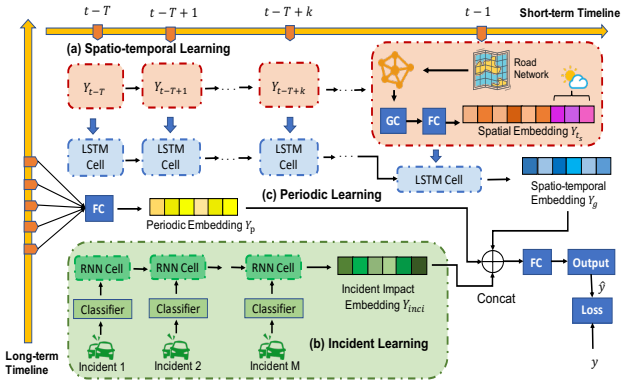


Figure 7: The architecture of DIGC-Net

6.1 Methodology

DIGC-Net (Figure 7) consists of three components: spatio-temporal learning, incident learning and periodic learning. Our prediction problem is defined in Section 3.

Spatio-temporal Learning (Figure 7(a)). Traffic speed prediction is also related to spatio-temporal patterns of traffic networks. Meanwhile, several previous works [22, 36, 42] use spatio-temporal features for traffic prediction (we also discuss them in Section 2). Therefore, we use the similar structure of spatial and temporal learning of our binary classifier. The spatial-temporal and context structure is a common use in traffic prediction, and we use GCN rather than CNN to better capture spatial features of road networks here. GCN is used for capturing spatial graph features and LSTM is adopted to capture the time evolution patterns of traffic speeds. The input features of each node is $V_{\xi_i}^t$ in GCN, i.e., the speed of each flow at time slot t. More specifically, the input features is $X^t = \left[V_{\xi_0}^t, V_{\xi_1}^t, \cdots, V_{\xi_{N-1}}^t\right]$, which is the graph speed snapshot at time slot t. We input a sequence of graph speed snapshots features $[X^{t-T}, X^{t-T+1}, \cdots, X^{t-1}]$ to GCN and after the GCN part, similar to [36], we concatenate the weather contexts at each time slot t to get Y_t. Then we feed the spatial features sequence $[Y_{t-T}, Y_{t-T+1}, \cdots, Y_{t-1}]$ to LSTM cells to iteratively get the output sequence $[h_{t-T}, h_{t-T+1}, \cdots, h_{t-1}]$. Then we use k learnable units to predict k future traffic speeds $\left[Y_S^t, Y_S^{t+1}, \cdots, Y_S^{t+k-1}\right]$. The output of spatio-temporal learning is Y_S.

Incident Learning (Figure 7(b)). To predict traffic speed at time slot t, we select all incidents occurred within $[t-125min, t-5min]$ as the incident learning inputs (the last two hours), where $t-125min$ is the earliest included incident occurrence time and $t-5min$ is the latest time. We use the pre-trained binary classifier (introduced in Section 5) to extract $(Y_c \oplus Y_g)_{FCs}$, i.e., the latent incident impact features of each incident. Because the number of incidents occur within the time range is uncertain and incidents occur in a sequential order, so we adopt standard Recurrent Neural Network (RNN) [25] for incident learning. RNN is a neural network that contains loops that allow information to persist. Previous incidents will affect the traffic conditions, which may lead to the occurrence of future incidents. Using RNN also help us capture the interrelation of sequentially occurring urban traffic incidents, which is neglected by previous works [19]. We denote Y_{inci} as the output of the last RNN cell.

Table 1: Evaluation of MAPE among different methods

Method	MAPE-SFO	MAPE-NYC
ARIMA	26.70 %	38.60 %
SVR	28.24 %	39.73 %
LSTM	18.98±0.18%	30.26±0.25%
GC	15.69±0.21%	25.79±0.32%
ConvLSTM	13.95±0.12%	22.80±0.18%
LSM-RN	13.72 %	21.53 %
STDN	13.45±0.12%	20.24±0.20%
LC-RNN	12.26±0.22%	18.77±0.36%
DIGC-Net	**11.02±0.15%**	**17.21±0.22%**

Periodic Learning (Figure 7(c)). Traffic flow speeds change periodically and we use the similar structure of [22] to learn long-term periodical patterns. We use the same time slots in the last 5 days to learn the periodic features. A fully connected layer is adopted to capture the long-term periodic features. The output of periodic learning is Y_P.

Output. After getting spatio-temporal features Y_S, incident impact features Y_{inci}, and periodic features Y_P, we adopt a concat operation to concatenate them, then feed them into m FC-layers. Finally we get the prediction value $\hat{y_t}$, and compute the loss compared with the real value y_t.

Objective Function and Evaluation Metric. DIGC-Net is training by minimizing the Mean Squared Error ($MSE = \sum_{i=1}^{N} (\hat{y_i} - y_i)^2$) between the predicted speed and the real value. Similarly to [7], we use Mean Absolute Percentage Error to evaluate DIGC-Net, MAPE is defined as : $MAPE = \frac{100\%}{N} \sum_{i=1}^{N} |\frac{\hat{y_i} - y_i}{y_i}|$, where N is the total number of flows.

6.2 Evaluations

Parameter Setting. The datasets we use here are listed in Section 3. We set 5 minutes as the time interval and time window as 4 hours, i.e., $T = 48$. We train our network with the following hyper-parameter settings: learning rate (0.001) with Adam optimizer. In spatio-temporal learning, we set two GCN layers followed by one FC-layer (64-dimension) and the input size of the first GCN layer is 64. We use *ReLU* activation function and add Dropout in GCN layer with $d = 0.5$. In incident learning, we use one RNN layer with 128-dimension hidden state. In periodical learning, we use one FC layer with 64-dimension hidden state. After the concat operation, we adopt one FC-layer with 256-dimension and connect the final output layer. We use *ReLU* activation function in the FC layers. We use first three weeks data for training and validation, and the remaining one week data as the test set. In training dataset, we select 90% of them for training and 10% as the validation set for early stopping.

Comparison with competitive benchmarks. We compare our model with the following models in consideration of covering

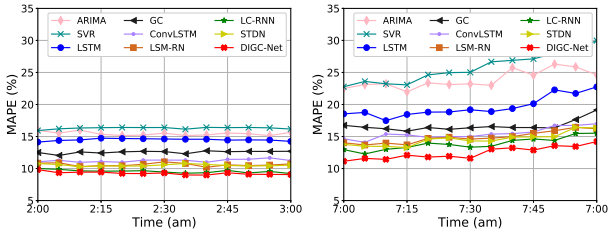
Table 2: Evaluation of MAPE among different variants of DIGC-Net

Variant	MAPE-SFO	MAPE-NYC
Spatio-temporal	12.47±0.11%	18.69±0.19%
Spatio-temporal + periodic	12.22±0.12%	18.63±0.19%
DIGC-Net-all (Spatio-temporal + periodic + incident)	**11.02±0.15%**	**17.21±0.22%**

widely used traditional machine learning, matrix decomposition and state-of-the-art deep learning methods: (1) Autoregressive integrated moving average (ARIMA) [5]; (2) Support Vector Regression (SVR) [31]; (3) LSTM [23]; (4) GC [6] is the variation of basic GCN with the efficient pooling; (5) ConvLSTM [30] extends the fully connected LSTM to have convolutional structures; (6) Latent space model for road networks (LSM-RN) [7] learns the attributes of vertices in latent spaces which mainly uses matrix decomposition; (7) LC-RNN [22] takes advantage of both RNN and CNN models and designs a Look-up operation to capture complex traffic evolution patterns, which outperforms ST-ResNet [43] and DCNN [24], so we do not further compare ST-ResNet and DCNN here; (8) STDN [37] uses a flow gating mechanism to explicitly model dynamic spatial similarity and uses a periodically shifted attention mechanism to capture temporal features.

Table 1 shows the MAPE results of using different methods of SFO and NYC. All other benchmarks in the table is one-step prediction. When compared with different methods, DIGC-Net achieves the best performance in both two cities. DIGC-Net has lower MAPE than these benchmarks in SFO (from 10.11% lower up to 60.97% lower) and lower MAPE than these benchmarks in NYC (from 8.31% lower up to 56.68% lower). We also note significant variance between SFO and NYC among all methods, likely due to large differences in the traffic road network (NYC is much larger than SFO: 2,416 vs 13,028 nodes and 19,334 vs 92,470 edges). The results indicate that DIGC-Net can effectively incorporate incident, spatio-temporal, periodic and context features for traffic speed prediction.

Comparison with variants of DIGC-Net. We also present the comparison with different variants of DIGC-Net with only spatio-temporal component, spatio-temporal and periodic component, and the whole DIGC-Net with all components (spatio-temporal, periodic and incident components). The comparison results are shown in Table 2. The first finding is that the performance improvement of periodic learning is relatively small, with only difference of 0.25% of SFO and 0.06% of NYC. One possible reason that the improvement margin of SFO is larger than NYC is that there is a relatively simple road network in SFO and the variation of traffic speed is more regular. The MAPE without incident learning (spatio-temporal + periodic) is 12.22% of SFO and 18.63 % of NYC, which also outperforms all benchmarks (sightly outperforms LC-RNN). It also verifies that our incident learning component is the key to the improvement with a 1.2% MAPE improvement of SFO and 1.42% MAPE improvement of NYC.



(a) Non-rush Hour (b) Rush Hour

Figure 8: Time-sensitive comparison of SFO

Table 3: Evaluation of MAPE for multi-step prediction

Method	MAPE-SFO	MAPE-NYC
DIGC-Net, k=1	**11.02±0.15%**	**17.21±0.22%**
DIGC-Net, k=2	11.36±0.19%	17.94±0.25%
DIGC-Net, k=3	11.62±0.26%	18.83±0.36%

Comparison with different time period. As shown in Figure 5(b) and Figure 5(d), the number of incidents varies over time, and more incidents occur at traffic peak periods. Meanwhile, traffic speed variation is also time-sensitive. Therefore, we further select 2:00 - 3:00 am as the non-rush hour and 07:00 - 08:00 am as the rush hour, and take SFO as the illustration to evaluate the performance of different methods. Figure 8 shows the MAPE results in the non-rush hour and rush hour. In the non-rush hour, our method has lower MAPE than these benchmarks in SFO (from 2.08% lower up to 64.43% lower), and lower MAPE than these benchmarks in the rush hour (from 10.78% lower up to 89.50% lower). The performance of our method and LC-RNN are similar in the non-rush hour but exhibit a relatively clear gap in the rush hour, which derives from more complex traffic patterns in the rush hour. Among them, we find LC-RNN is better than STDN, and we believe it is because that the look-up operation of LC-RNN can extract spatial features more effectively than common convolution in the non-Euclidean network structures.

Comparison of multi-step prediction. We then present the comparison results for multi-step prediction. DIGC-Net can be used for multi-step speed prediction by setting k learnable units in spatio-temporal learning component. We set prediction length $k = 1, 2, 3$ (speeds of next 5, 10 and 15 minutes) to evaluate the multi-step prediction case. The results are shown in Table 3. The performance of DIGC-Net of multi-step prediction remains stable as the predicted length increases (drop relatively 3.09% of $k = 2$ and 5.44 % of $k = 3$ compared with $k = 1$ in SFO and drop relatively 3.88% of $k = 2$ and 9.03% of $k = 3$ compared with $k = 1$ in NYC). When prediction length is within three steps, DIGC-Net outperforms all other baselines of one-step prediction in SFO, and in NYC, only of one-step that LC-RNN outperforms three-steps DIGC-Net. The multi-step results demonstrate that our model can be effectively applied to multi-step prediction within a certain time range.

7 CONCLUSION

In this work, we investigate the problem of incident-driven traffic speed prediction. We first propose a critical incident discovery method to identify urban crucial incidents and their impact on traffic flows. Then we design a binary classifier to extract the latent incident impact features for improving traffic speed prediction. Combining both processes, we propose a Deep Incident-Aware Graph Convolutional Network (DIGC-Net) to effectively incorporate traffic incident, spatio-temporal, periodic and weather features for traffic speed prediction. We evaluate DIGC-Net using two real-world urban traffic datasets of large cities (SFO and NYC). The results demonstrate the superior performance of DIGC-Net and validate the effectiveness of latent incident features in our framework.

ACKNOWLEDGMENTS

This work has been sponsored by National Natural Science Foundation of China (No. 61602122, No. 71731004, No. 61971145), CERNET Innovation Project (NGII20190105), the project "PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019)", Academy of Finland under grant number 317432 and 318937. Yang Chen is the corresponding author.

REFERENCES

[1] HERE Traffic API. 2019. *https://developer.here.com/.*
[2] K. Boriboonsomsin et al. 2012. Eco-routing navigation system based on multi-source historical and real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems* 13, 4 (2012), 1694–1704.
[3] J. Bruna et al. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR' 14).*
[4] M. Castro-Neto et al. 2009. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications* 36, 3 (2009), 6164–6173.
[5] J. Contreras et al. 2003. ARIMA models to predict next-day electricity prices. *IEEE Transactions on Power Systems* 18, 3 (2003), 1014–1020.
[6] M. Defferrard et al. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 29th Neural Information Processing Systems. (NIPS '16).*
[7] D. Deng et al. 2016. Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16).*
[8] I. S. Dhillon et al. 2004. Kernel K-means: spectral clustering and normalized cuts. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Dining (KDD '04).*
[9] R. Gao et al. 2019. Aggressive driving saves more time? multi-task learning for customized travel time estimation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI '19).*
[10] Y. Gu et al. 2016. From twitter to detector: real-time traffic incident detection using social media data. *Transportation Research Part C: Emerging Technologies.* 67 (2016), 321–342.
[11] Y. He et al. 2019. Traffic influence degree of urban traffic accident based on speed ratio. *Journal of Highway and Transportation Research and Development (English Edition)* 13, 3 (2019), 96–102.
[12] S. Hochreiter and J. Schmidhuber. 1997. Long Short-term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[13] R. J. Javid and R. J. Javid. 2018. A framework for travel time variability analysis using urban traffic incident data. *IATSS Research* 42, 1 (2018), 30–38.
[14] I. Johnson et al. 2017. Beautiful… but at what cost?: an examination of externalities in geographic vehicle routing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–21.
[15] T. N. Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of 5th International Conference on Learning Representations (ICLR '17).*
[16] Y. Li et al. 2018. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In *Proceedings of the 6th International Conference on Learning Representations. (ICLR '18).*
[17] Z. Li et al. 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. *IEEE Transactions on Intelligent Transportation Systems* 18, 11 (2017), 3204–3217.
[18] L. Lin et al. 2015. Modeling the impacts of inclement weather on freeway traffic speed: exploratory study with social media data. *Transportation Research Record* 2482, 1 (2015), 82–89.
[19] L. Lin et al. 2017. Road traffic speed prediction: a probabilistic model fusing multi-source data. *IEEE Transactions on Knowledge and Data Engineering* 30, 7 (2017), 1310–1323.
[20] L. I. Lin. 1989. A concordance correlation coefficient to evaluate reproducibility. *Biometrics.* 67 (1989), 255–268.
[21] Y. Lv et al. 2014. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 865–873.
[22] Z. Lv et al. 2018. LC-RNN: a deep learning model for traffic speed prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18).*
[23] X. Ma et al. 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54 (2015), 187–197.
[24] X. Ma et al. 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17, 4 (2017), 818.
[25] T. Mikolov et al. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH'10).*
[26] M. Miller and C. Gupta. 2012. Mining traffic incidents to forecast impact. In *Proceedings of the 1st ACM SIGKDD International Workshop on Urban Computing (Urbcomp '12).*
[27] B. Pan et al. 2012. Utilizing real-world transportation data for accurate traffic prediction. In *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM '12).*
[28] M. M. Rathore et al. 2016. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks* 101 (2016), 63–80.
[29] D. J. Rumsey. 2015. U can: statistics for dummies. (2015).
[30] X. Shi et al. 2015. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In *Proceedings of Neural Information Processing Systems (NIPS' 15).* 802–810.
[31] A. J. Smola and B. Schölkopf. 2004. A tutorial on support vector regression. 14, 3 (2004), 199–222.
[32] Y. Tong et al. 2017. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD' 17).*
[33] N. Viovy et al. 1992. The best index slope extraction (BISE): a method for reducing noise in NDVI time-series. *International Journal of Remote Sensing* 13, 8 (1992), 1585–1590.
[34] R. Xie et al. 2018. We know your preferences in new cities: mining and modeling the behavior of travelers. *IEEE Communications Magazine* 56, 11 (2018), 28–35.
[35] Yahoo. 2019. *https://developer.yahoo.com/weather/.*
[36] H. Yao et al. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI' 18).*
[37] H. Yao et al. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI' 19).*
[38] B. Yu et al. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI' 17).*
[39] S. X. Yu and J. Shi. 2003. Multiclass spectral clustering. In *Proceedings the 9th IEEE International Conference on Computer Vision (ICCV '03).*
[40] Z. Yuan et al. 2018. 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 and Data Mining (KDD '18).*
[41] H. Zhang et al. 2018. Detecting urban anomalies using multiple Spatio-temporal data sources. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 54:1–54:18.
[42] J. Zhang et al. 2016. DNN-based prediction model for spatial-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. (SIGSPATIAL '16).*
[43] J. Zhang et al. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI '17).*
[44] C. Zheng et al. 2020. Gman: a graph multi-attention network for traffic prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI' 20).*
[45] J. Zheng and L. M. Ni. 2013. Time-dependent trajectory regression on road networks via multi-task learning. In *Proceedings of 27th AAAI Conference on Artificial Intelligence (AAAI' 13).*