Australasian Transport Research Forum 2021 Proceedings
8-10 December, Brisbane, Australia
Publication website: http://www.atrf.info

Data-driven Traffic Incident Prediction with Hybrid Graph-based Neural Network

Thanh Tran[1], Dan He[1], Jiwon Kim[1] and Mark Hickman[1]

[1]School of Civil Engineering, The University of Queensland, St. Lucia, QLD

Email for correspondence: jiwon.kim@uq.edu.au

1. Introduction

Traffic incident management plays an essential role in Intelligent Transportation Systems since incidents such as vehicle crashes usually cause severe congestion on traffic networks and even human fatalities. Accurate incident prediction that estimates the probability of whether a traffic incident will happen or not in a specific region ahead of time would be helpful to provide road safety guidance and improve traffic conditions by preventing congestion. However, it is challenging to achieve acceptable prediction performance since traffic incidents could be caused by multiples factors such as traffic condition, weather condition, road structures, and driver behaviours, which are often difficult to capture in a given prediction model due to a lack of data or a challenge in fusing heterogeneous data sources. Recently, with the ubiquitous availability of location-aware sensor technologies such as GPS devices, more diverse sets of traffic data have become available in addition to traditional loop detector data. The increase in diverse traffic data can facilitate traffic prediction solutions, but a new challenge arises in the fusion of data from multiple sources with different granularity and penetration. As such, data-driven approaches that can flexibly leverage diverse data sources are highly desirable. In the literature, many classical machine learning methods are applied in traffic incident prediction such as K-nearest neighbor (Lv, Tang, and Zhao 2009) and Bayesian Network (Hossain and Muromachi 2012), which make predictions based on manually generated features extracted from traffic incident data. The generalisation ability of these algorithms is relatively low since important factors causing incidents including traffic flow data, road networks, and meteorological data are not considered. Recently, deep learning techniques have been applied to predict traffic incidents integrating more data sources. Particularly, Ren et al. (2018) developed a model base on Long Short-Term Memory (LSTM) to predict traffic incident risk by learning periodical temporal patterns and regional spatial correlation, and the Hetero-Convolutional LSTM model (Yuan, Zhou, and Yang 2018) learns from the inputs as flattening vectors extracted from the images of the map. But no topological information of the underlying traffic network is captured by these models. Later, Yu et al. (2021) handled this problem by proposing a graph-based model to learn spatial-temporal, external features from a graph that represents a road network. However, most of the existing work fail to consider data fusion from multiple sources to enhance the model performance.

In this paper, we propose a Data-driven Hybrid Graph-based Neural Network (DHGNN), which aims to predict the likelihood of traffic incidents within a given region ahead of a certain time period. There are three major contributions to our work. (1) **All-in-One**: Unlike most of the existing work that one model can be applied to only one specific region or sub-network of the whole traffic network, our solution is able to predict the incident occurrence for different city-wide sub-networks by one model. Specifically, we randomly sample thousands of sub-networks from one studied traffic network as the underlying input graphs with different structures. Features are extracted regarding multiple factors for each sub-network. By learning the spatial and temporal correlations (Table 1) of incident/nonincident cases occurring in different other sub-networks, our model can predict whether a given sub-network has an incident or not ahead of time. (2) **Data Integration:** Heterogeneous data from different sources relevant to traffic incidents, including traffic data (flow, occupancy, speed), network structures, and temporal information are integrated and normalised to form the features for different sub-networks. Our data are captured by two different types of sensors—loop detectors and probe vehicles—and, thus, the data granularity and coverage vary widely. In order to deal with the data sparsity problem in one dataset to match the density of the other, we apply the *K-Pod clustering* technique (J. T. Chi, E. C. Chi, and Baraniuk 2016) to figure out the representative samples to fill the missing data points based on similarity. (3) **Hybrid Neural Networks:** Since our two datasets cover different roads in the same region, we must construct different sub-networks to form input cases. However, a single graph neural network is insufficient to deal with one sample with two different graph structures. Thus, we propose a hybrid graph neural network that contains two sub-modules to embed two samples with different graph structures, followed by a general fully connected layer to output the prediction result. As a result, our model can achieve superior performance with 92.8% accuracy and 92.5% in AUC, which will be presented in detail in Section 4. Our model is flexible in terms of the extension to integrate other data sources by augmenting other corresponding neural networks.

2. Data Preparation

For output data, we use incident data from Queensland, Australia in 2017, where we mainly focus on *vehicle crashes* among various incident types. For input traffic data, we use two different data sources: *STREAMS* (n.d.) and *HERE* (n.d.) from Queensland, Australia in 2017. STREAMS dataset contain traffic flow and occupancy collected from loop detectors, while HERE dataset contain speed captured by GPS probes. While both are link-level measures, they use different link systems and network representations. We normalise the data in 5-minute aggregation and extract features for these two datasets separately: 16 features from STREAMS, $\boldsymbol{X^s}$=$< x_1^s, x_2^s, \ldots, x_{16}^s >$ and 8 features from HERE, $\boldsymbol{X^h}$=$< x_1^{\mathrm{h}}, x_2^{\mathrm{h}}, \ldots, x_8^{\mathrm{h}} >$, as summarised in Table 1.

Table 1: Description of Features from STREAMS and HERE Data

Notation	Feature description	Notation	Feature description
x_1^s, x_2^s, x_3^s	$\mathrm{p}_{t-2}, \mathrm{p}_{t-1}, \mathrm{p}_t$	x_4^s, x_5^s, x_6^s	$rp_{t-2}, rp_{t-1}, rp_t \left(rp_t = \frac{p_t - \overrightarrow{p_t}}{sp_t} \right)$
x_7^s, x_8^s, x_9^s	f_{t-2}, f_{t-1}, f_t	$x_{10}^s, x_{11}^s, x_{12}^s$	$rf_{t-2}, rf_{t-1}, rf_t \left(rf_t = \frac{f_t - \overrightarrow{f_t}}{sf_t} \right)$
x_{13}^s	30-minute flow ratio	x_{14}^s	free-flow speed
x_{15}^s	length of link (STREAMS)	x_{16}^s	level of service (STREAMS)
x_1^h, x_2^h, x_3^h	$\mathrm{v}_{t-2}, \mathrm{v}_{t-1}, \mathrm{v}_t$	x_4^h, x_5^h, x_6^h	$r\mathrm{v}_{t-2}, r\mathrm{v}_{t-1}, r\mathrm{v}_t \left(r\mathrm{v}_t = \frac{\mathrm{v}_t - \overrightarrow{\mathrm{v}_t}}{s\mathrm{v}_t} \right)$
x_7^h	speed limit (HERE)	x_8^h	length of link (HERE)

- t: 5-minute time interval;
- $\mathrm{p}_t/f_t/\mathrm{v}_t$: aggregated occupancy/flow/speed at *t*;
- $\overrightarrow{\mathrm{p}_t}/\overrightarrow{f_t}/\overrightarrow{\mathrm{v}_t}, \mathrm{sp}_t/sf_t/\mathrm{sv}_t$: are historical mean and standard deviation of occupancy/flow/speed at t;
- flow ratio: the increasing/decreasing trend across the previous 6 time slots $\frac{1}{6}\sum_{t=2}^{6}(f_t - f_{t-1})$

Rather than studying the link-level incident prediction, we focus on *sub-networks*. Specifically, we randomly sample thousands of regions with 500-m radius in Brisbane and, for each region, we construct two sub-networks representing traffic data from the abovementioned two sources, each of which contains its own set of road links with features. One challenge in using the HERE data we have was a missing data problem, where a lot of links had no speed observation for a specific time period. To address this issue, we apply the K-Pod clustering technique to capture the representative pattern of speed for each sub-network for a given time period and use it to replace any missing data.

3. DHGNN Model

In this section, we introduce the overall framework of our model. Figure 1 shows the architecture of DHGNN, which is comprised of three major components: input module, sub-network embedding module, and output module.

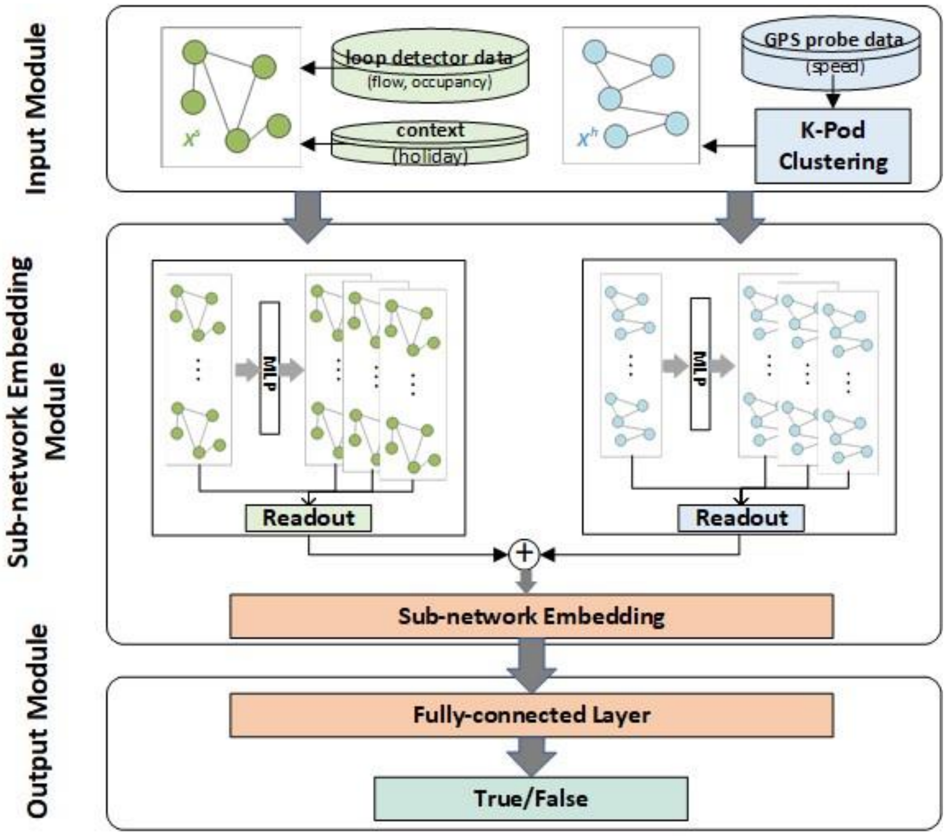


Figure 1: The overview of Hybrid Graph-based Neural Network

Input Module. As illustrated on top of Figure 1, we prepare the input data for our model by integrating multiple data sources using the approach introduced in Section 2. Then, for each case (incident/non-incident), we have two sub-networks with different graph structures but representing the same geographical region. Note that the nodes in the graph represent road links and edges indicate the connectivity among links. We attach two sub-networks with two different groups of features X^s and X^h respectively.

Sub-network Embedding Module. Next, in the sub-network embedding module, there are two graph neural networks (GNNs) taking the two featured sub-networks as input, respectively. It is expected that incidents happening on one link could be affected by traffic conditions or other factors from its nearby links. Generally, GNNs capture the dependence of graphs following a neighbourhood aggregation strategy, in which the massages of a node are iteratively updated by aggregating massages of its neighbours. After k iterations of aggregation, one node's massages then contain the structural information within its k-hop network neighbourhood. Thus, the correlations of multiple factors that influence traffic incidents in the sub-networks can be learnt by such models. In our work, we apply the ChebyGIN (Knyazev, Taylor, and Amer

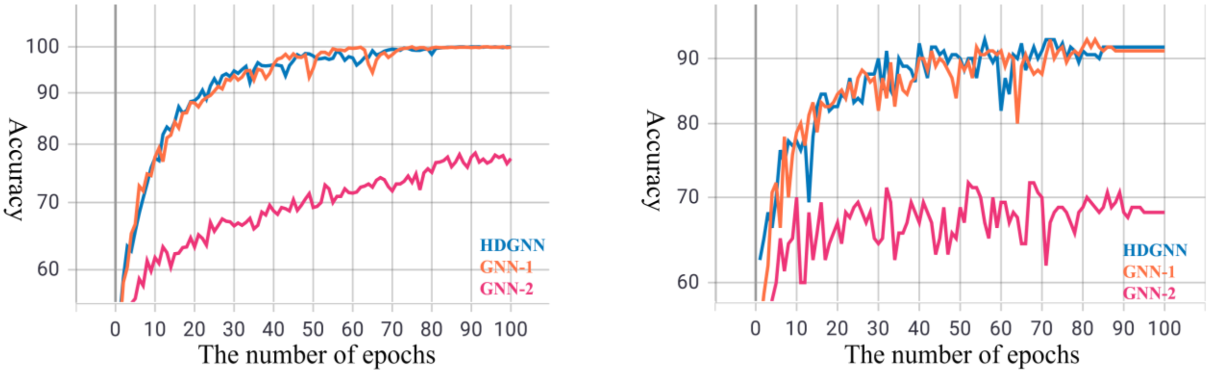
2019) model, which is a variation of the Graph Isomorphism Network (GIN) Xu et al. (2019). Intuitively, the underlying network structure is an important factor for the occurrence of traffic incidents. Compared to other GNNs, the GIN model is capable to learn the difference of graph structures from different sub-networks. As shown in Figure 1, the input sub-networks are fed into the initial graph convolution layer, where features in the nodes are propagating to their neighbours. Afterwards, before the next iteration of message passing, multi-layer perceptrons (MLPs) is used as the composition of aggregation functions. In the readout stage, where the node-level massages are summarised to generate the final embedding capturing all the information from the entire graph, rather than making use of only the final iteration, we use information from all iterations of the model to consider all the structural information. In the end, the readouts from two ChebyGIN models are concatenated to generate the embedding of two sub-networks.

Output Module. After the sub-network embedding, we apply a fully connected layer to generate the final prediction result. The output of the prediction result is either True or False representing there exists incident or non-incident within the given region ahead of time.

4. Experiments

4.1. Experimental settings

In our experiment, our model is evaluated by four metrics: *Accuracy*, *AUC* (Area Under the Curve), *Precision* and *Recall*. Regarding the prediction horizon, we make use of up to 30 minutes of historical traffic data to predict whether there will be an incident in the following 15 minutes. The traffic incident predicted by our model are mainly vehicle crash. Since traffic incidents are rare events, we adopt the undersampling method to balance the cases for incident and non-incident. There are 800 incident cases from one-year data, and we randomly sample another 800 non-incident cases from the same year. For model training, we split our dataset into three parts: 70%,10%,20% for training, validating, testing, respectively. We evaluate the performance of our model compared to two baseline models that contain only one ChebyGIN making use of loop detector & context data and GPS probe data respectively, denoted by GNN-1 (using loop detector & context data only) and GNN-2 (using GPS probe data only).



(a) Training **(b) Validation**

Figure 2: Training and Validation Performance

4.2. Prediction Performance

Figure 2 shows the accuracy during training (a) and validation (b) in DHGNN, GNN-1 and GNN-2. The performance tends to be stable when the number of epochs is around 80, showing the successful learning of our model. Table 2 reports the evaluation metrics of our DHGNN, GNN-1, GNN-2. The performance of DHGNN is better than that of GNN-1 and GNN-2, indicating the importance of using all the information from loop detector, context, and GPS probe data, which was possible through our model's ability to integrate multiple data sources with different link representations and graph structures.

Table 2: Performance of Models on Test Set

Models	Accuracy	AUC	Precision	Recall	Accuracy: (TP+TN)/(TP+TN+FP+FN) Precision: TP/(TP+FP) Recall: TP/(TP+FN) TP:True positive; TN:True negative; FP:False positive; FN:False negative
DHGNN	**0.928**	**0.925**	**0.910**	**0.950**	
GNN-1	0.911	0.919	0.892	0.938	
GNN-2	0.712	0.724	0.732	0.668	

5. Conclusion

In this paper, we propose a deep learning model, named DHGNN, for traffic incident prediction, which contains two GNNs sub-modules to learn the correlations of traffic data from multiple sources. Experimental results show the superiority of our hybrid model compared to the one with one GNN making use of a single data source. In the future, we will extend our model by augmenting another neural network sub-module integrating more data sources, e.g., weather data.

6. Acknowledgements

The authors gratefully acknowledge the support of Australian Research Council (ARC), Queensland Department of Transport and Main Roads, and Transmax Pty Ltd under the ARC Linkage Project on Real-time Analytics for Traffic Management (LP180100018).

Reference

Chi, Jocelyn T., Eric C. Chi, and Richard G. Baraniuk (2016). "*k*-POD: A Method for *k*-Means Clustering of Missing Data". In: *The American Statistician* 70, pp. 91–99.

HERE (n.d.). url: https://developer.here.com/products/platform.

Hossain, Moinul and Yasunori Muromachi (2012). "A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways". In: *Accident Analysis & Prevention* 45, pp. 373–381.

Knyazev, Boris, Graham W Taylor, and Mohamed Amer (2019). "Understanding attention and generalization in graph neural networks". In: *NIPS*, pp. 4202–4212.

Lv, Yisheng, Shuming Tang, and Hongxia Zhao (2009). "Real-time highway traffic accident prediction based on the k-nearest neighbor method". In: *ICMTMA*. Vol. 3. IEEE, pp. 547–550.

Ren, Honglei et al. (2018). *A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction*.

STREAMS (n.d.). url: https://www.transmax.com.au/what-we-do/streams/.

Xu, Keyulu et al. (2019). "How Powerful are Graph Neural Networks?" In: *ICLR*.

Yu, Le et al. (2021). "Deep spatio-temporal graph convolutional network for traffic accident prediction". In: *Neurocomputing* 423, pp. 135–147.

Yuan, Zhuoning, Xun Zhou, and Tianbao Yang (2018). "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data". In: *SIGKDD*, pp. 984–992.