Leveraging 3D-Raster-Images and DeepCNN with Multi-source Urban Sensing Data for Traffic Congestion Prediction

Ngoc-Thanh Nguyen[1,2], Minh-Son Dao*[3], and Koji Zettsu[3]

[1] Vietnam National University, Ho Chi Minh City, Vietnam
[2] University of Information Technology, Ho Chi Minh City, Vietnam
thanhnn.13@grad.uit.edu.vn
[3] National Institute of Information and Communications Technology, Japan
dao,zettsu@nict.gp.jp

Abstract. Nowadays, heavy traffic congestion has become an emerging challenge in major cities, which should be tackled urgently. Building an effective traffic congestion predictive system would alleviate its impacts. Since the transit of vehicles heavily depends on its spatial-temporal correlations and effects of exogenous factors such as rain and accidents, they should be simultaneously considered. This study proposes a deep learning approach based on 3D-CNN to utilize many urban sensing data sources wrapped into 3D-Raster-Images. Armed with this, the spatial and temporal dependencies of the data can be entirely preserved. Furthermore, traffic congestion status of different geographical scales at various time horizons can be fully explored and analyzed. We also propose data fusion techniques to (1) fuse many environmental factors that affect vehicles' movements, and (2) incorporate social networking data to improve predictive performance further. The experiments are performed using a dataset containing four sources of urban sensing data collected in Kobe City, Japan, from 2014-2015. The results show that the predictive accuracy of our models improves significantly when using multiple urban sensing data sources. Finally, to encourage further research, we publish the source code of this study at `https://github.com/thanhnn-uit-13/Fusion-3DCNN-Traffic-congestion`.

Keywords: Traffic congestion · Deep learning · Spatio-temporal data

1 Introduction

In [1], Lana et al. argued that traffic congestion prediction, which is a part of the urban computing domain, is a crucial and urgent topic. The authors also introduced many exciting new topics such as concept drift in data-driven models, big data, and architecture implementation. Finally, they pointed out that increasing the prediction time horizon (i.e., longer than 1 hour) and incorporating exogenous factors to models are two of the major challenges.

* corresponding author

In [2], the authors introduced a fascinating overview of fusing urban big data based on deep learning. This research emphasized the critical role of urban big data fusion based on deep learning to get more values for urban computing.

In [3], the vital role of traffic data visualization was seriously concerned. The authors compiled several techniques designed for visualizing data based on time, location, spatial-temporal information, and other properties in traffic data. They aim to satisfy the need to have good traffic data visualization to analyze as well as reveal hidden patterns, distributions, and structure of the data.

In light of the above discussions, we introduce a new deep learning and multi-source data fusion approach to predict the average traffic congestion length from different road segments concurrently appearing in the predefined area by using multiple urban sensing data sources with variable time-horizons. To do this, we first wrap spatial-temporal information of multi-source data into 2D/3D raster-images by using the method introduced in [4]. Then, we utilize recent advances in convolutional neural networks to build a deep learning model for traffic congestion prediction. The inputs of the model are (1) environmental sensing data that are fed via separate channels and (2) social networking data which is incorporated with the channels based on the content of the posts via a weighted function. The techniques significantly enhance the prediction performance of our models. The contribution of this research is summarized as follows:

- **Multi-scale Time Horizons**: [1] argued that short-term traffic (60 minutes or below) had been a big interest of the research community, but the investigations on medium- and long-term are neglected. Some more recently published works like [5], [6], [7], and [8] also did not address it. We explore all the time windows mentioned. The predictive time ranges from 1.5 hours to 1 day with multiple immediate steps varying from 30 minutes to 4 hours.
- **Multi-source/Multimodal Big Data Fusion**: This study proposes data-fusion techniques to simultaneously extract and feed to models various urban sensing data types including (1) traffic congestion, (2) precipitation, (3) vehicle collisions, and (4) social networking data. While the first three categories were included in many highly influential works such as [5][6], the last one has attracted less interest and even required more research to conclude its impact[9]. Our proposed models success them by considerably improving the predictive performance compared to models using single-source data.
- **Spatial-temporal Wrapping**: The raster-image-based wrapping technique utilized in this study can reserve the correlations between spatial and temporal information, which are crucial in urban-related domains. It also allows us to look back and give predictions over variable time horizons with multiple steps unlike [5], [10] which give predictions only on single time steps. Last but not least, by using this data format, we can leverage sophisticated 2D-CNN and 3D-CNN layers to build learning models.

The paper is organized as follows: Section 2 describes procedures to pre-process multi-source data. Section 3 explains fusion techniques to merge many sources of sensing data. Section 4 discusses contributions of the proposed method. Finally, Section 5 concludes the work and suggests possible future research directions.

2 Data Preprocessing

This section discusses the procedure to prepare data that is fed to models.

2.1 Data storage

This study uses four different urban data sensing sources:

- Traffic congestion: offered by Japan Road Traffic Information Center, JARTIC (www.jartic.or.jp),
- Precipitation: received from Data Integration and Analysis System Program, DIAS (www.diasjp.net),
- Vehicle collisions: got from Institute for Traffic Accident Research and Analysis, ITARDA (www.itarda.or.jp), and
- Social networking data: bought from TWITTER (www.twitter.com). Twitter provided us with posts that contain certain keywords relating to bad traffic status, heavy rain, and vehicle collisions in the requested/examined area. Each post contains *username*, *location*, *timestamp*, *concerned_keywords* information. In this study, this data is referred to as "SNS".

The mentioned types of sensing data are stored in a data warehouse system called Event Warehouse (EvWH) introduced in [4]. The table *analysis.raw_transation* stores four different urban sensing values at each local area identified by *meshcode* per timestamp t. Each location has a size of 250m x 250m.

2.2 Converting urban sensing data to raster-images

This section presents the procedure to convert spatiotemporal urban sensing data stored in time-series format to 2D multi-layer raster-images. We store the four urban sensing sources to a single 2D multi-layer raster-image per timestamp t. Using this data format, we can simultaneously analyze their effects at a specific location at any time t given. For example, Figure 1 illustrates how (1) traffic congestion (Blue channel), (2) rain (Green channel), and (3) traffic accidents (Red channel) affected Osaka Bay, Kobe City, Japan at 10:00:00 AM July 17th, 2015. Their effects are well-interpreted via a single RGB raster-image. Since this problem deals with multiple time steps, we place many raster-images consecutively to form a 3D raster-image. Thus, it helps reserve the temporal correlations of the data.

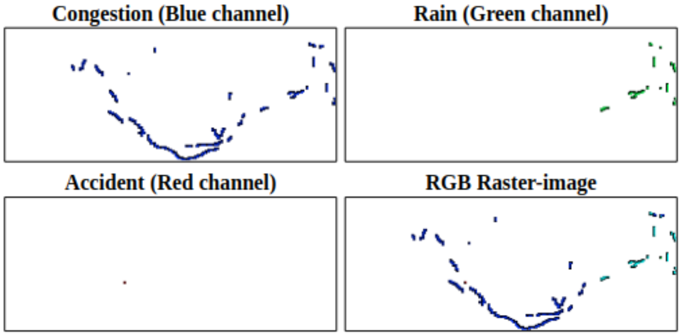

Fig. 1. Leveraging raster-images to visualize urban data

In this study, we examine the area of 20km × 50km, which is represented on a raster-image sized 80 × 200 ($W \times H$). It is illustrated in Figure 2. Each pixel on a raster-image pixel equals to a local area sized 250m x 250m. Green points denote locations that have traffic congestion in the ground truth data. We use four different sensing data types (L), so each produced raster-image has a size of 80 x 200 x 4 ($W \times H \times L$). Each sensing event is triggered and recorded every 5 minutes, so 288 raster-images are generated per day. For traffic prediction, we divide the examined area further into three smaller regions as denoted in red squares in Figure 2. Since the other areas are waters or have very little traffic information, ignoring them will significantly alleviate the impact of sparse data.

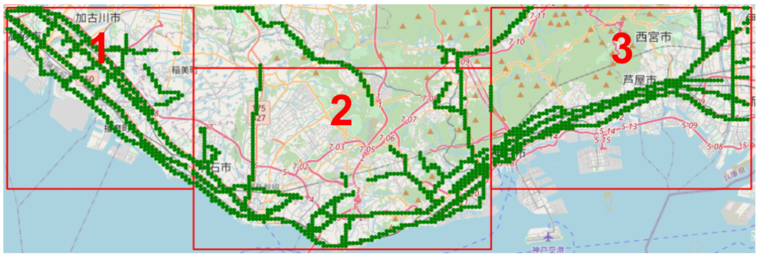

Fig. 2. Region for traffic congestion prediction

Next, we explain the detailed procedure to convert time-series sensing data to 3D multi-layer raster-images in Algorithm 1. Following is the configurations of 3D multi-layer raster-images:

- Global map's size: $H \times W$ (Height x Width);
- Number of sensing data types: L (Layer);
- Top-left coordinate of the map: $base\_lat$, $base\_lon$;
- Delta between 2 adjacent geographical locations: d_lat, d_lon;
- Number of time steps considered: S.

Algorithm 1 Convert multi-source sensing data to a 3D multi-layer raster-image

```
    Input: Time-series sensing data: $data$
    Output: A 3D multi-layer raster-image $R$, sized $S \times H \times W \times L$.

Initialize a raster-image $R$
while $s \in S$ do
    Read sensing data at time $t + s$ into $data\_ts$
    while $l \in data\_ts$ do
        //Extract relative position on the raster-image
        $loc\_x \leftarrow (loc\_lat - base\_lat)/l\_lat$
        $loc\_y \leftarrow (loc\_lon - base\_lon)/l\_lon$

        //Assign sensing data to the desired location
        $R[s, loc\_x, loc\_y, l] \leftarrow l\_sensingdata$
    end while
end while
```

3 Learning model

3.1 Leveraging Computer Vision's breakthroughs

After completing the procedures explained in Section 2, the data is stored in 3D multi-layer raster-images. They are similar to a video of conventional RGB pictures, so we can utilize research breakthroughs of the Computer Vision domain. [2] indicated that spatial and temporal relationships should be simultaneously considered when analyzing traffic movements. In [11], Tran et al. proposed 3D Convolutional Neural Network (3D-CNN) which can fully reserve such dependencies. Therefore, we decide to utilize this network in our learning models.

To better explain the complete solution of the study, we illustrate the whole workflow in Figure 3. Firstly, the urban sensing data sources are converted to 3D Multi-layer Raster-Images by Algorithm 1 in Section 2. Then, different layers of the 3D multi-layer raster-images are either fed to our proposed learning models (called **Fusion-3DCNN**) in separate channels or integrated with others. We will explain them shortly. The input and output of Fusion-3DCNN are 3D Single-layer Raster-Images. The models look back k ($k \geq 1$) historical milestones of multiple environmental factors to predict m periods ($m \geq 1$) of traffic congestion. Depending on the values of k and m, we will flexibly stack the number of 2D multi-layer raster-images to form a 3D multi-layer raster-image accordingly. This technique allows us to give predictions for various immediate time steps in different time horizons.

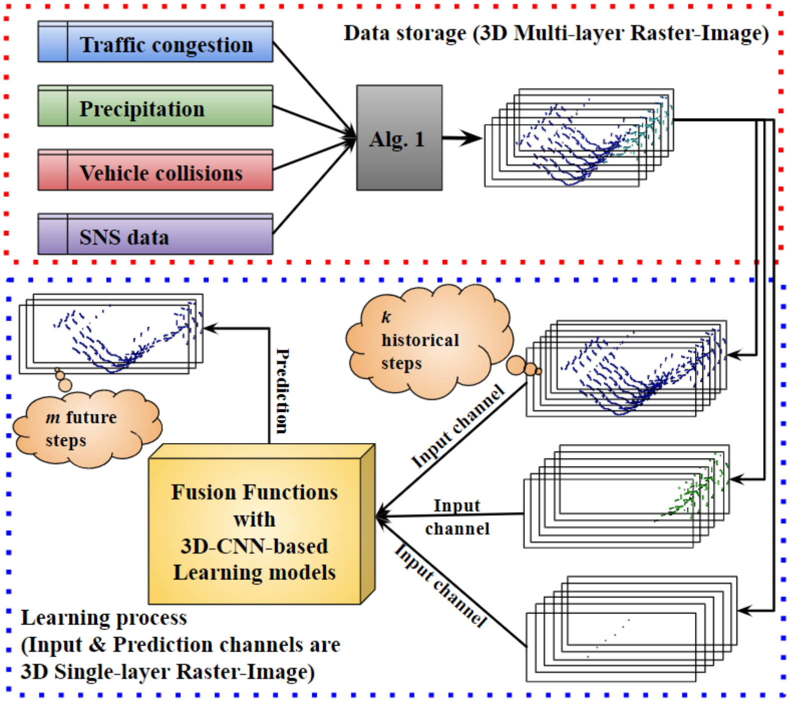
Data storage (3D Multi-layer Raster-Image)
Traffic congestion
Precipitation
Vehicle collisions
SNS data
Alg. 1
k historical steps
m future steps
Prediction
Input channel
Input channel
Input channel
Fusion Functions with 3D-CNN-based Learning models
Learning process (Input & Prediction channels are 3D Single-layer Raster-Image)

Fig. 3. Complete workflow of the study

3.2 Data fusion techniques

This section reveals fusion techniques to feed multiple data sources to Fusion-3DCNN. The traffic may be affected by two types of sensing data: environmental factors and social networking data. The former contains traffic congestion, precipitation, and vehicle collisions. They are called "environmental factors". They are learnable features and directly supplied to Fusion-3DCNN as follows: $W \times x = \sum_{i=1}^{N} W_{f_i} \times x_{f_i}$; where f_i denotes individually learnable factors. The second sensing data group includes Tweets of Twitter's users complaining/warning about the bad surrounding environment. There are three types of criticism: (1) heavy traffic congestion, (2) heavy rain, and (3) vehicle collisions. As a matter of fact, if one is warned about bad environmental conditions at a location in his commuting route, he would avoid reaching it. Therefore, the negative effects of these factors on traffic congestion would be mitigated. The following formula denotes how large each explicit factor changes based on the community's online activities: $x^f_{(i,j)} = \begin{cases} (1-p) \times x^f_{(i,j)}, & \text{if } y^f_{(i,j)} = 1 \\ x^f_{(i,j)}, & \text{otherwise} \end{cases}$ where $x^f_{(i,j)}$ is the normalized value of the affected factor type f at the location (i,j) on raster-images; $y^f_{(i,j)}$ is a binary value indicating whether a warning related to that factor is detected (1: yes / 0: no); and p is a hyperparameter defining impact level of the SNS data on environmental factors.

3.3 Building predicting models

This section discusses some detailed information of Fusion-3DCNN as follows:

1. Fusion-3DCNN simultaneously considers historical data of many urban sensing data sources to predict future traffic congestion in the examined geographical areas. The model receives k 3D single-layer raster-images (equivalent to k milestones) to produce m 3D single-layer raster-images containing m future traffic congestion situations.
2. Fusion-3DCNN predicts traffic congestion status for separate geographical areas, which are marked in red in Figure 2. The biggest examined area indicated by 2 has the size of 60×80 on a raster-image, so the smaller regions will be padded with 0s to compensate for the sizes which are different. All areas are fed into learning models simultaneously during the training process.

The architecture of Fusion-3DCNN is illustrated on the main page of the work's repository[4] with details about filter size and activation functions. The models are implemented in Keras with Tensorflow backend. All 3D-CNN layers have kernel size is (3, 3, 3) and are set to same padding, one-step striding. The models are optimized with Adam optimizer with MSE loss function. They are trained on Geforce GTX 750Ti GPU with 2GB VRAM. The batch size is 1, the learning rate is 3e-5 to 5e-5, and decayed by 1e-5 to 2e-5. The sensing data of 05-10/2014 is used for training, and the data of 05-10/2015 is utilized for testing.

[4] `https://github.com/thanhnn-uit-13/Fusion-3DCNN-Traffic-congestion`

4 Empirical Evaluation

4.1 Evaluative preparation

This section discusses settings prepared to evaluate models. Firstly, we prepare three distinct datasets with different looking back and predicting time horizons, as in Table 1. Next, we use four baselines namely (1) Historical Average - HA, (2) Sequence-to-Sequence Long Short-Term Memory - Seq2Seq LSTM, (3) 2D Convolutional Neural Network - 2D-CNN, and (4) 3D Convolutional Neural Network - 3D-CNN. They only use traffic congestion data. The first three neglect either spatial or temporal dependencies, while the last one reserves both. Therefore, 3D-CNN-based models are expected to be the best baseline. Beside, to show the effectiveness of using multi-source data, Fusion-3DCNN models are expected to be better than all the baselines. To evaluate the models' performance, we use Mean absolute error (MAE). Subsequently is the baselines' information:

Table 1. Experimental datasets

Data type description	**Dataset type (-term)**		
	Short	**Medium**	**Long**
No. looking back frames	6	6	6
Delta between looking back frames	30 min.	1 hr.	4 hr.
Total looking back time	3 hr.	6 hr.	24 hr.
No. prediction frames	3	3	6
Delta between prediction frames	30 min.	1 hr.	4 hr.
Total prediction time	1.5 hr.	3 hr.	24 hr.

- **HA:** traffic congestion value at each geographical area is calculated by: $C^f_{i,j} = \frac{1}{N}\sum_{h=1}^{N} C^h_{i,j}$; where $C^h_{i,j}$ and $C^f_{i,j}$ represents traffic congestion value at the location (i, j) on raster-images at historical milestones h and future stages f, respectively. This model only reserves temporal information.
- **Seq2Seq LSTM:** this network has achieved great successes on different tasks relating to sequential data[12]. However, it only learns temporal dependencies. We use five ([400 × 5]) hidden LSTM layers for both encoder and decoder components, and train in 300 epochs.
- **2D-CNN:** this network is very efficient in learning and predicting data that requires the reservation of spatial information[13]. However, it totally neglects the temporal dimension. We use eight hidden 2D-CNN layers $[[128\times 2]-[256\times 4]-[128\times 1]-[64\times 1]]$ in this baseline, and train in 1 epoch.
- **3D-CNN:** this network can reserve both spatial and temporal dimensions[11]. We use eight hidden layers $[[128\times 2]-[256\times 4]-[128\times 1]-[64\times 1]]$, and train in 1 epoch.

Next, the models that use multi-source data are prepared as follows:

1. One Fusion-3DCNN model that uses (1) traffic congestion, (2) precipitation, and (3) vehicle collisions; and
2. Two Fusion-3DCNN models that gather the above three factors and social networking posts (SNS data). In the experiment, we perform a grid search to see how much the SNS data affects the other factors. Two impact levels are evaluated in this study: $[25\%, 75\%]$.

4.2 Evaluative results discussion

This section discusses conclusions extracted from empirical evaluation. Models' performance are presented in Table 3. Better models have lower values. Insights concluded from the results are also discussed subsequently. Table 2 identifies models presented in Table 3 with shorter names to reduce the space consumed.

Table 2. Shortened model names

Model	**Shortened name**
Historical Average	HA
Seq2Seq LSTM	Seq2Seq
2D-CNN	2D-CNN
3D-CNN using Congestion	3D-CNN with C
Fusion-3DCNN using Congestion & Precipitation & Accidents	Fusion-3DCNN CPA
Fusion-3DCNN using Congestion & Precipitation & Accidents & Reduce their 25% impacts by SNS data	Fusion-3DCNN CPA*¼SNS
Fusion-3DCNN using Congestion Precipitation & Accidents & Reduce their 75% impacts by SNS data	Fusion-3DCNN CPA*¾SNS

Table 3. Experimental results (measured in MAE)

Dataset (-term)	**Model**						
	HA	**Seq2Seq**	**2D-CNN**	**3D-CNN with C**	**Fusion-3DCNN CPA**	**Fusion-3DCNN CPA*¼SNS**	**Fusion-3DCNN CPA*¾SNS**
Short	5.51	54.47	5.37	5.19	5.04	**4.90**	4.98
Medium	6.60	52.05	6.45	6.31	6.12	5.70	**5.64**
Long	7.05	56.33	6.85	6.63	6.16	5.86	**5.72**

Firstly, by considering the results of baselines which only use traffic congestion data, some conclusions can be drawn as follows:

- The predictive performance of Seq2Seq models in all datasets are quite bad compared to the others. That is because there is too much sparse data in our dataset, which accounts for 95%. Alvin et al. indicated that LSTM-based models are data-hungry models[14], so this argument explains our problem. Because of that, we will ignore this baseline and only analyze the other baselines. Furthermore, deep learning models that use 2D-CNN and 3D-CNN show acceptable performance (better than the naive baseline - HA). It indicates that convolutions of those networks can effectively tackle data sparsity problems.
- The predictive performance of 2D-CNN models are slightly better than HA in all datasets. Besides, the predictive accuracy of 3D-CNN models are higher than the other baselines. It indicates that spatial and temporal dependencies are crucial. Therefore, leveraging 3D-CNN layers in building Fusion-3DCNN models is a correct choice in this study.

Next, some observations can be extracted when comparing the predictive performance 3D-CNN and Fusion-3DCNN models as follows:

- Generally, the predictive performance of Fusion-3DCNN models considering multiple sources of data are better than that of 3D-CNN models that use only traffic congestion data in all datasets. The predictive accuracy of Fusion-3DCNN CPA models are higher than 3D-CNN models by 3% to 7%. The gaps even extend when compare Fusion-3DCNN CPA-SNS with 3D-CNN. Particularly, the predictive accuracy of Fusion-3DCNN CPA-SNS models are higher than 3D-CNN by 4% to 14%. It indicates that all external environmental factors considered in this study really affect traffic congestion. It also proves that our proposed data fusion functions and strategies to leverage many urban sensing data sources simultaneously is effective in enhancing the predictive performance. Moreover, incorporating SNS data helps further increase forecasting accuracy. Therefore, this study has opened a promising research direction to utilize this abundant and easy-to-collect source of information in the studies relating to traffic congestion prediction.

- Considering the short-term dataset, once leveraging rainfall and traffic accident data, the predictive performance of Fusion-3DCNN CPA increases by 3% compared to 3D-CNN. Incorporating SNS data to the environmental factors makes the predictive performance of Fusion-3DCNN CPA*¼SNS higher than Fusion-3DCNN CPA by 3%. Besides, since the predictive accuracy of Fusion-3DCNN CPA*¾SNS is lower than Fusion-3DCNN CPA*¼SNS, it indicates that aggressively reducing environmental impacts via SNS data could make models perform poorly. Therefore, it could be concluded that a minority of the population (about 25%) tend to update the latest environmental status during their short-distance commute.
- Moving to the longer-term datasets (medium-term and long-term), external factors have a more significant impact on traffic congestion compared to the short-term time window. In more detail, Fusion-3DCNN CPA models are better than 3D-CNN by 5% and 8% for medium-term and long-term datasets, respectively. It shows that when heavy rain or a traffic accident occurs, they are likely to affect traffic flows on longer time horizons (3 hours-24 hours versus 1 hour 30 minutes). In addition, the best models in these two datasets are Fusion-3DCNN CPA*¾SNS. The predictive performance of these models are better than Fusion-3DCNN CPA by 8% in both the datasets. The results reveal that a majority of people try to catch up with the latest happenings when planning their travel in these longer forecasting time horizons.

5 Conclusions

This study proposes a deep learning approach to predict the traffic congestion status of each geographical area in the examined regions by reserving both spatial and temporal information. It also simultaneously considers external factors that can affect the flows of vehicles. We successfully show the positive impacts of using environmental factors such as rain, traffic accident, and social networking contents in enhancing predictive performance. Owing to the advances of modern-day technology, collecting a variety of urban sensing data and social networking information is feasible. Thus, this work has opened a promising research direction to utilize various additional factors besides traffic congestion data. Finally, our raster-image-based wrapping solution could be utilized to integrate more urban data types perfectly. They play a vital role in building an ideal learning model to predict traffic congestion.

References

1. I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2):93–109, 2018.
2. J. Liu, T. Li, P. Xie, S. Du, F. Teng, and X. Yang. Urban big data fusion based on deep learning: An overview. *Information Fusion*, 53:123 – 133, 2020.
3. W. Chen, F. Guo, and F. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015.
4. M. Dao and K. Zettsu. Complex event analysis of urban environmental data based on Deep-CNN of spatiotemporal raster images. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2160–2169. IEEE, 2018.
5. X. Yuan, Z.and Zhou and 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 '18, pages 984–992, 2018.
6. F. Tseng, J. Hsueh, C. Tseng, Y. Yang, H. Chao, and L. Chou. Congestion prediction with big data for real-time highway traffic. *IEEE Access*, 6, 2018.
7. M. Chen, X. Yu, and Y. Liu. PCNN: Deep Convolutional Networks for Short-Term Traffic Congestion Prediction. *IEEE Trans. on Intelligent Transportation Systems*, 19(11):3550–3559, 2018.
8. Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1720–1730, 2019.
9. N. Pourebrahim, S. Sultana, J. Thill, and S. Mohanty. Enhancing trip distribution prediction with twitter data: Comparison of neural network and gravity models. In *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, GeoAI'18, pages 5–8, 2018.
10. X. Di, Y. Xiao, C. Zhu, Y. Deng, Q. Zhao, and W. Rao. Traffic congestion prediction by spatiotemporal propagation patterns. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pages 298–303, June 2019.
11. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA, 2015. IEEE Computer Society.
12. T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. Reddy. Neural abstractive text summarization with sequence-to-sequence models. *arXiv preprint arXiv:1812.02303*, 2018.
13. W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11 – 26, 2017.
14. A. Kennardi and J. Plested. Evaluation on neural network models for video-based stress recognition. In *Neural Information Processing*, pages 440–447, Cham, 2019. Springer International Publishing.