

Early Identification of Recurrent Congestion in Heterogeneous Urban Traffic

Lin Zhu, Rajesh Krishnan, Fangce Guo, John W. Polak and Aruna Sivakumar

Abstract—Urban traffic congestion has become a critical issue that not only affects the quality of daily lives but also harms the environment and economy. Traffic patterns are recurrent in nature, so is congestion. However, little attention has been paid to the development of methods that would enable early warning of the formation of congestion and its propagation. This paper proposes a method for automated early congestion detection operating over time horizons ranging from half an hour to three hours. The method uses a deep learning technique, Convolutional Neural Networks (CNN), and adapts it to the specific context of urban roads. Empirical results are reported from a busy traffic corridor in the city of Bath. Comprehensive evaluation metrics, including Detection Rate, False Positive Rate and Mean Time to Detection, are used to evaluate the performance of the proposed method compared to more conventional machine learning methods including Feed-forward Neural Network and Random Forest. The results indicate that recurrent congestion can indeed be predicted before it occurs and demonstrates that CNN based method offers superior detection accuracy compared to the conventional machine learning methods in this context.

I. INTRODUCTION

Traffic congestion has significant impacts on the lives of people in urban areas [1]. Generally, the types of traffic congestion can be divided into two categories: Recurrent Congestion (RC) and Non-Recurrent Congestion (NRC) [2], [3]. RC refers to a day-to-day congestion where traffic demand exceeds the road capacity. RC usually occurs at peak periods; commuters and network operators can anticipate RC in advance based on the historical traffic profiles [4]. With adequate advance notice, one can make plans to mitigate RC and implement suitable intervention measures. On the other hand, NRC is usually caused by unpredictable and unexpected events (e.g., traffic incidents and accidents), planned events (e.g., concerts, football matches and road works) and inclement weather. These events can occur at any time of day and location and event duration usually depends on the conditions of the road network [5]. NRC cannot be accurately predicted in advance. Urban traffic congestion poses a wide range of challenges that not only affect the commuters' daily lives but also result in negative impacts on the economy, environment and society [6]. Therefore, it is beneficial to quickly diagnose RC onset and accurately detect congestion evolution patterns in order to anticipate traffic bottlenecks and proactively develop strategies to mitigate them.

Recurrent traffic congestion has been extensively studied for decades because of the growing necessity to manage urban road networks more efficiently [7], [8], [9]. However, to date, there are two major challenging problems that remain unsolved. One is with a complex and constantly changing traffic demand profile, how to improve the accuracy of traffic congestion detection by considering the spatio-temporal propagation of RC. The other one is any possibility to detect the onset of RC before its occurrence or given early stage warning. Such understanding can provide valuable information for traffic engineers to effectively predict the traffic state and mitigate congestion in advance.

The aim of this paper is therefore to develop a deep learning based detection method that can provide accurate early detection of RCs on urban roads. In order to achieve this aim, three objectives are identified. First, we will develop a deep learning based RC detection method that can capture spatial and temporal correlations of traffic congestion data on urban roads. The second is to detect RCs as early as possible in order to give an early alarm. The third is to determine criteria to evaluate different RC early detection methods in terms of two aspects, i.e., number of time windows (i.e., how many prior time steps should be used as inputs), number of time steps ahead (how far it can early detect before congestion). The proposed early detection method is tested using traffic flow and occupancy data from the City of Bath.

The rest of this paper is organised as follows. Section II briefly reviews the relevant literature on RC detection and discusses the motivation of the proposed approach. Section III describes the proposed RC early detection method and the proposed criteria to evaluate the performance of RC detection methods. Section IV describes the data sources used in this paper and presents the evaluated results and comparison analysis. Conclusion and recommendations are drawn and discussed in Section V.

II. LITERATURE REVIEW

The existing literature provides various urban RC prediction methods and techniques that have been employed to examine traffic data on urban roads. In this paper, we present an overview of recent advances of data-driven based methods, i.e., deep learning techniques. Traffic congestion has previously been characterised as a clustering problem in terms of both theoretical and empirical aspects. Theoretical studies investigate the movement of vehicles at critical traffic densities on motorways [10]. The empirical research focuses on making use of the spatio-temporal cluster to detect congestion patterns [2].

Traditional RC prediction methods analyse the collected traffic data with the aim of predicting or estimating traffic variables, such as traffic flow, speed and travel time, in a short-time window. Since these variables are essential indexes of Level of Service (LoS) and congestion, over the past decades, research efforts have resulted in a substantial number of prediction methods [11], [12] and several surveys have summarised these methods [13], [14]. However, to date, most of the previous studies tend to view congestion in a short-term time window, i.e., 5-minute or 15-minute time window. This short-term period may be insufficient for traffic operators to proactively develop management plans and cover the latency of varies communication. Little attention has been paid, however, to the development of methods that would enable the early detection of the formation of congestion and the characteristics of its spatio-temporal evolution. In a broad sense, the analysis of time window size is vital for determining how early an incident could be detected. Hence, a more straightforward RC early detection method which can infer to the onset of RC directly will largely lessen the time-constraint burden of traffic operators and provide them with an RC index for proactive reaction.

Over the last few years, most research studies have attempted to predict or estimate traffic congestion on motorways or highways (e.g., [15] and [16]). Nevertheless, the application of RC detection on the context of urban roads is less studied because of its complex traffic composition with irregular interruptions (e.g., intersections, traffic lights, pedestrian crossings and on-street parking) [2]. Therefore, new techniques are required to investigate the detection of RC in urban areas. In this paper, we propose the use of a deep learning method to address the gaps mentioned above. This proposed method is aimed to be used for automated RC early detection and alerting of road recurrent congestion.

The phrase *deep learning* refers to machine of techniques that use multiprocessing layers to learn the relationship within the data by using backpropagation algorithms [17]. Recently, considerable efforts focused on the application of deep learning to traffic congestion problem. For example, [18] used a deep Restricted Boltzmann Machine and Recurrent Neural Networks to predict traffic congestion based on GPS data collected from taxis. [19] used an error-feedback recurrent CNN structure to predict continuous traffic speed as its ability to learn from prediction errors so as to adapt for abrupt traffic anomalies. Recently, [20] applied a CNN method to learn the traffic as an image with the aim to predict large-scale and network-wide traffic speed taking account of spatio-temporal traffic dynamics with relatively high accuracy. Deep learning methods, especially CNN, can construct much deeper and more sophisticated architectures than a conventional machine learning method and they can therefore directly capture the spatio-temporal traffic features and correlations in the urban traffic network as a whole on a large scale network [20].

CNN is a branch of deep learning techniques initially introduced in 1980 [21], as derivatives of conventional multilayer neural networks. CNN has been widely and

successfully used in many fields with remarkable results, such as image processing [22] and signal processing [23]. In addition to the fully connected layers found in conventional multilayers, a CNN includes convolutional layers and pooling layers where the convoluted layers are locally connected and parameters are significantly reduced in the pooling process. The locally-connected convoluted layers enable a CNN to capture complex spatial correlation problems [22], while reduce parameters in the pooling layer which makes a CNN potentially applicable to large-scale traffic network problems [24]. CNN has been used extensively in applications involving complex spatio-temporal data and its ability to deal with spatial structure suggests that it has the potential to be used in the RC early detection problem.

III. METHODOLOGY

This paper proposes a novel method for the automated early detection and alerting of recurrent traffic congestion, operating over time windows ranging from half an hour to three hours and early detect the congestion states before 1 to 8 time steps ahead where each time step stands for 15 minutes. The whole methodology procedure consists of two steps, namely (1) label generation and (2) early detection (see Fig. 1).

A. Label Generation

In the first step, an unsupervised learning method, the Expectation Maximisation (EM) algorithm, is used to classify (“generate labels” in the parlance of the machine learning community) traffic states into non-congestion state and congestion state, i.e., a set of $\{0, 1\}$, by using the ratio of traffic occupancy o to flow q . Two assumptions have been made in this paper: firstly, traffic states are clustered into two separate regimes by either non-congestion state or congestion state; secondly, the probability distribution of these traffic states follows a Gaussian distribution with mean μ_i and variance σ_i^2 , where i refers to traffic states $\{0, 1\}$. Then, a Gaussian mixture model is formulated by

$$X \sim \gamma_0 \mathcal{N}(\mu_0, \sigma_0^2) + \gamma_1 \mathcal{N}(\mu_1, \sigma_1^2) \quad (1)$$

with the constraint of

$$\gamma_0 + \gamma_1 = 1 \quad (2)$$

With Gaussian distribution of these traffic states, the question is then modelled to solve the probability

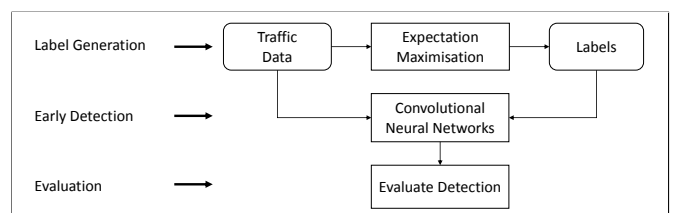


Fig. 1: The methodology framework

of each traffic state with unknown parameters set $(\sigma_0, \sigma_1, \mu_0, \gamma_0^2, \mu_1, \gamma_1^2)$. EM algorithm is used to find maximum likelihood estimates of parameters with an iteration of Expectation (E) step which creates an expectation of the log-likelihood for the parameters and Maximisation (M) step, which maximises the log-likelihood in the previous step. [25] gave more details of the formulated Gaussian mixture distribution and EM algorithm.

B. Early Detection

EM algorithm can be viewed as a form of unsupervised learning method in which this context is used to generate labels that can in turn be used to drive a more sophisticated CNN-based supervised deep learning method, which is used to detect the early onset of congestion. In the second step, these labels are used as the pre-defined labels to a CNN-based early detection model. During this step, time-series traffic flow data will be converted into a 3D time-space feature where x-axis, y-axis and z-axis represent time, space and size of time windows respectively as CNN inputs. Each cell inside the matrix represents the rate of occupancy to flow in each time step. The 3D time-space input of CNN is shown in Fig. 2.

After converting into a 3D time-space feature space, we define the architecture of the CNN model to extract spatial features of 3D time-space matrix. Every layer in a CNN model serves as a detection filter where the first layer recognises the relatively obvious features; the following layers gradually detect the more abstracted features while the last layer of a CNN makes a specific classification based on all features detected by previous layers. The architecture of the proposed CNN is described as follows.

The convolutional layer is the most important layer in a CNN model [22]. It connects the input matrix defined as $x \in \mathcal{R}^{p \times t \times s \times w}$ with a set of filters $W \in \mathcal{R}^{n \times p \times m \times n}$ where p is the sample size, $t \times s \times w$ is the size of the input matrix corresponding to time, space and size of time windows respectively, and n is the number of convolutional filters while $m \times n$ is the size of convolutional kernel. The single convolutional layer is formulated as

$$f(x, W, b) = \{ReLU(x \otimes W_k + b_k)\}_{k=1, \dots, n} \quad (3)$$

where $b \in \mathcal{R}^n$ is a bias for each filter output and $\otimes$ is the convolution operator that applies on a single input x and

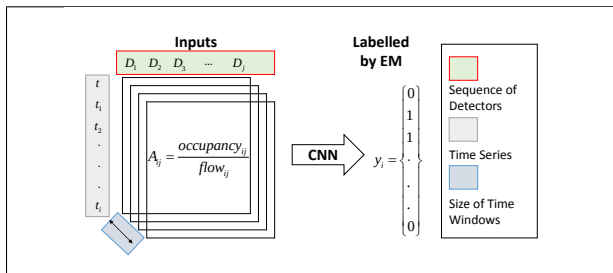


Fig. 2: 3D time-space inputs of CNN

k th filter. *ReLU* refers to Rectified Linear Unit, which is a non-saturating activation function as derived in Eq. 4 that can capture the nonlinearities of the output [26].

$$f(x) = \max(0, x) \quad (4)$$

The most common pooling strategy, a max pooling technique, is applied after the convolutional layer to output the maximum. The fully-connected layer is employed in the last stage of hidden layers to control the dimension of final output. A binary sigmoid crossentropy loss $\mathcal{L}$ function is used as an objective function to be minimised later with the predicted value $\hat{y}_i$ and ground truth value y_i .

$$\mathcal{L} = -\sum_i \hat{y}_i \log(y_i) - (1 - \hat{y}_i) \log(1 - y_i) \quad (5)$$

RMSprop [27] with updated weight and bias in each iteration serves as an optimiser to utilise the magnitude of recent gradients in order to minimise the loss function. The parameters are updated as follows.

$$r_t = \gamma f'(\theta_{t-1})^2 + (1 - \gamma) f'(\theta_t)^2 \quad (6)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{r_t + \epsilon}} f'(\theta_t) \quad (7)$$

where $f'(\theta_t)$ is the derivative of the loss function $\mathcal{L}$ with respect to the parameters θ at time step t and ϵ is an error term.

IV. RESULTS AND ANALYSIS

In this section, the proposed method was tested with real-world case study, i.e., the City of Bath. With the labels generated from EM algorithms, the results classified by the CNN model were compared to other traditional machine learning based on several performance indicators such as false positive rate and detection rate. Furthermore, we investigated the robustness of the algorithms and its effectiveness in detecting traffic congestion in terms of the size of time windows and detection horizons.

A. Data Description

The proposed detection methods are tested using traffic flow and occupancy data from the City of Bath. The case study is based on the data from a corridor in the city centre, Lower Bristol Road, shown in Fig. 3, with 18 Inductive Loop Detectors (ILDs). The traffic data covers two years from 1st May 2015 to 31st May 2017 in 15-minute time intervals with a total sample size of 35424. The data has been divided with 70% for training, 10% for validation and 20% for testing. The Fig. 4 shows the statistical properties of the dataset. As expected, the traffic flow at weekends are generally fewer than the weekdays, while in terms of time of day, the traffic flow during peak hours are fewer than that of off-peak hours.

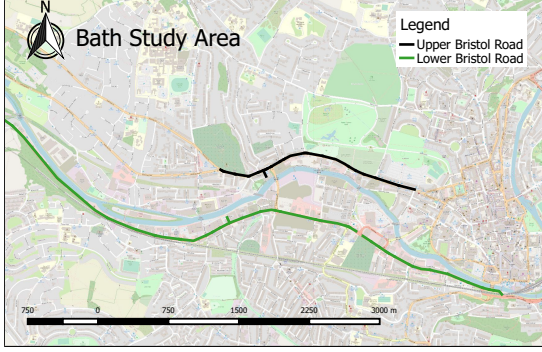


Fig. 3: Lower Bristol Road in Bath

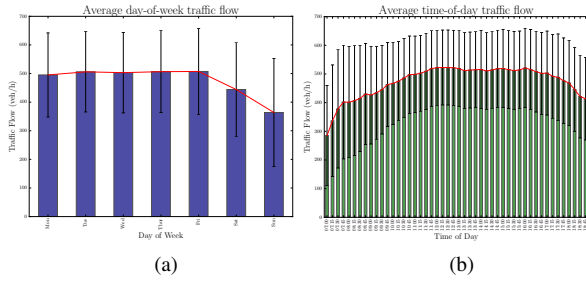


Fig. 4: The average traffic flow of: (a) day of week; and (b) time of day

B. EM Algorithm

The experiment will start with pre-defined binary labels by using the EM algorithm. EM algorithm has been proved to be an effective and transferable probabilistic traffic state classifier which can capture the latent features of the underlying distribution [25]. With its inherent advantage of generating labels automatically, EM may work well with any recurrent congestion detection algorithms without labelling traffic states manually. One EM binary cluster result is shown in Fig. 5, where the EM algorithm classifies the traffic states into two regimes based on traffic occupancy and traffic flow.

The time-series data have been transformed to 3D matrix, where x -axis represents 48 time steps of each day ranging from 7am to 6:55pm, y -axis represents the sequence of 18 detectors, ordered spatially with reference to a predefined

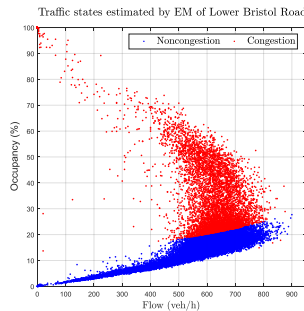


Fig. 5: Performance of EM algorithm based on traffic states clustered by the EM algorithm

start point of the selected road, and z -axis stands for the size of time windows varying from 1 to 8 time lags.

C. CNN Model

After labelling the congestion states (i.e., CNN targeted outputs) and transforming the time-series data into 3D matrix (i.e., CNN inputs), the proposed CNN is trained and fine-tuned. There are two types of parameters considered when implementing the structure of a CNN:

- the hyper-parameters connected with each layer and
- the depth of the CNN

As for hyperparameters, the max pooling size 2×2 is selected in this paper because it is a typical pooling kernel shape to effectively reduce the size of feature maps, and the kernel size of 3×3 and 2×2 are used for the proposed convolutional layers as its benefits of a decrease in the number of parameters and multiple applications of ReLU layers [28].

The depth of CNN is constrained by the input size ($48 \times 18 \times \text{size of time windows}$) of the image and thus the final net contains 5 learned layers including 3 convolutional and 2 fully-connected layers. Fig. 6 shows one example of CNN architecture with hyperparameters.

D. Other Machine Learning Competitors

Other conventional machine learning methods, such as Multilayer Perceptron (MLP) and Random Forest (RF) are used to compare with the proposed CNN method in the early detection part.

MLP is inspired by emulating the structure of biological networks and consists of neurons and multiple layers, i.e., input layer, hidden layer and output layer. It is designed to address complex, linear or nonlinear problem where the interrelationships among inputs are not well defined and understood [29].

RF uses a combination of classification tree instead of a single tree to detect the incident. RF trains many decision tree classifiers to construct the classifier ensemble, and then the ensemble is used to detect the incidents. RF has been used in previous studies and proved to have accurate incident detection performance compared with normal decision tree classifier [30].

MLP and RF, which have the good capability of capturing the complex and non-linear relationship between different features, are compared with the proposed CNN. Stochastic gradient methods are used to minimise the loss function

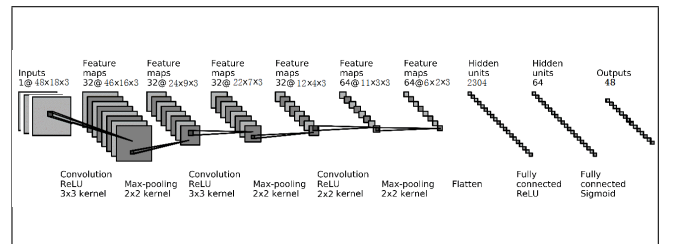


Fig. 6: Convolutional Neural Network architecture

and update the weights and bias step by step. After the optimisation, these methods with optimised weights are used to compare with the proposed CNN model.

As a result, MLP is set up with a hidden layers size of (10, 2) and regularisation parameter of 0.00001; RF is configured to generate 10 decision tree with the depth of 3. As for the data input for machine learning methods above, the time series flow data can be directly input into the models without the matrix transformation.

E. Results and Sensitivity Analysis

In order to comprehensively evaluate the performance of the proposed model, two evaluation indexes are also used in this paper. They are False Positive Rate (FPR) and Detection Rate (DR). FPR and DR are calculated using Eq. 8 and Eq. 9.

$$FPR = \frac{FP}{TN+FP} \quad (8)$$

$$DR = \frac{TP}{FN+TP} \quad (9)$$

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

The result of recurrent congestion early detection are presented in Table I, Table II and Table III for CNN, MLP and RF respectively. It is notable that both proposed CNN model and conventional machine learning techniques perform well in the early detection problem. However, we can find that the CNN marginally outperformed other machine learning methods in terms of high DR. Even though the CNN algorithm may need a longer training time than other competitors because of the constraints of inherent deep structure of the CNN, it exhibits acceptable computation time to early detection compared to the duration of time step ahead (i.e., 15 mins).

TABLE I: Performance of CNN for Early Detection

Horizons	Window Size							
	0	1	2	3	4	5	6	7
DR(%)								
0-step	94.7	96.7	97.4	98.1	95.0	91.2	95.5	90.8
1-step	97.2	95.0	94.5	94.3	96.7	93.9	90.2	97.2
2-step	97.5	94.9	93.0	94.6	98.1	96.8	93.3	89.0
3-step	97.9	96.5	96.9	92.3	87.9	89.9	87.3	95.6
4-step	94.5	90.3	91.6	89.6	91.7	93.0	90.0	92.6
5-step	94.9	93.3	97.7	95.0	96.6	91.5	88.7	86.7
6-step	94.8	91.3	98.4	91.4	95.1	94.6	88.5	97.6
7-step	97.0	94.6	91.9	85.6	99.0	93.0	92.8	99.2
FPR(%)								
0-step	2.0	2.3	2.2	2.0	2.2	2.4	2.3	2.2
1-step	1.9	2.3	2.4	2.1	2.4	2.3	2.3	2.1
2-step	2.2	2.4	2.4	2.3	2.0	2.2	2.1	2.2
3-step	0.9	2.1	2.2	2.1	2.3	2.2	2.3	2.1
4-step	1.7	2.5	2.3	2.3	2.5	2.3	2.4	2.2
5-step	2.0	2.5	2.1	2.2	2.5	2.5	2.3	2.2
6-step	1.8	2.6	1.9	2.3	2.2	2.2	2.2	2.1
7-step	1.2	2.3	2.6	2.5	2.0	2.3	2.5	1.7

TABLE II: Performance of MLP for Early Detection

Horizons	Window Size							
	0	1	2	3	4	5	6	7
DR(%)								
0-step	90.6	96.0	93.2	92.8	95.3	93.8	95.1	93.3
1-step	90.6	96.0	93.7	95.3	94.9	92.7	96.1	92.7
2-step	90.6	96.1	92.5	94.6	95.6	92.0	95.4	93.9
3-step	90.8	96.3	94.0	95.2	94.7	92.0	94.1	93.1
4-step	91.1	96.3	94.4	92.1	95.4	92.5	93.4	93.4
5-step	90.6	96.1	93.7	95.1	95.4	92.1	92.5	92.8
6-step	90.6	95.9	93.2	95.1	95.3	91.8	93.9	92.2
7-step	94.5	96.0	93.5	94.6	95.2	92.0	93.7	92.6
FPR(%)								
0-step	2.9	1.2	2.2	2.3	1.5	2.0	1.6	2.1
1-step	2.9	1.2	2.0	1.6	1.7	2.3	1.3	2.3
2-step	2.9	1.2	2.4	1.8	1.4	2.5	1.5	1.9
3-step	2.8	1.1	1.9	1.6	1.7	2.5	1.9	2.2
4-step	2.7	1.1	1.8	2.6	1.5	2.3	2.1	2.1
5-step	2.9	1.2	2.0	1.6	1.5	2.5	2.4	2.3
6-step	2.9	1.3	2.2	1.6	1.5	2.5	2.0	2.4
7-step	1.7	1.2	2.1	1.8	1.6	2.5	2.0	2.3

TABLE III: Performance of RF for Early Detection

Horizons	Window Size							
	0	1	2	3	4	5	6	7
DR(%)								
0-step	66.4	41.2	10.6	0.0	14.0	0.6	6.2	52.8
1-step	49.7	51.2	52.8	71.5	0.7	42.5	28.8	0.8
2-step	82.0	9.8	2.5	35.6	5.8	0.0	6.8	31.0
3-step	60.1	92.9	63.6	28.5	62.4	49.1	13.3	20.6
4-step	85.8	40.7	5.7	41.7	66.7	11.3	9.3	0.0
5-step	77.0	59.1	75.8	62.0	33.3	6.8	27.6	41.1
6-step	89.4	36.3	1.1	18.3	45.8	0.3	13.0	0.1
7-step	89.0	36.4	23.9	53.2	40.4	13.5	4.4	0.0
FPR(%)								
0-step	9.6	15.5	22.4	24.9	22.2	24.1	23.4	13.1
1-step	13.7	13.2	13.3	8.6	24.7	15.5	18.9	24.0
2-step	5.4	21.9	24.0	17.6	23.7	24.2	23.3	18.0
3-step	11.2	2.2	10.5	19.1	11.1	14.0	22.1	20.2
4-step	4.3	15.6	23.4	16.2	9.9	22.1	22.9	24.1
5-step	6.8	11.3	7.3	11.2	18.1	23.0	19.1	15.8
6-step	3.2	16.6	24.2	21.3	15.2	24.2	22.1	24.1
7-step	3.4	16.6	19.8	13.4	16.5	21.7	23.8	24.1

In order to measure the impact of time spans and early detection horizons, we conducted a sensitivity analysis in terms of size of time windows and time steps ahead respectively. As the results indicated in the Fig. 7, the sensitivity of three methods based on DR and FPR are varied significantly. CNN performs best with time window size of 2 – 3 and 0 – 1 time ahead early detection, while MLP can achieve better results with the time window size of 1 but it is insensitive to the number of time steps ahead. RF has the best performance with time window size of 1 and 3 time steps ahead detection and its performance will reduce with the size of time windows increase (see Fig. 7).

V. CONCLUSION

In this paper, an early detection model based on deep learning is presented to detect the early evidence of traffic congestion. The time-space relationship is captured during

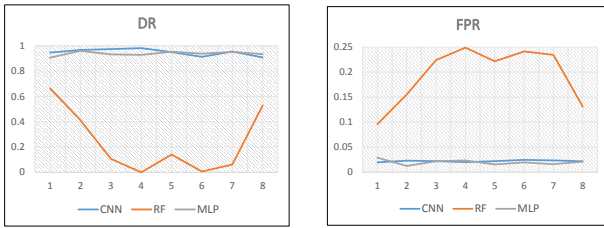


Fig. 7: Sensitivity analysis in terms of size of time windows

the feature transformation process in the deep learning neural layers. Comprehensive evaluation metrics, including the DR and FPR, are used to evaluate the performance of the proposed CNN compared to more conventional machine learning methods, i.e., MLP and RF. Comparison of the CNN method with established methods indicates that the deep neural network methods such as the CNN algorithm may be capable of achieving an accurate early recurrent congestion detection and outperforming conventional detection methods. The sensitivity of size of time windows and time steps is varied among three methods, while MLP is insensitive to the number of time steps ahead.

Future studies will design more experiments with multiple traffic state labels to examine the performance of the deep learning based early detection model. The model could be further validated using other real traffic data by considering the quality of data sources.

VI. ACKNOWLEDGEMENTS

The authors would like to thank Martin Wylie at Bath City Council for providing the traffic detector data used in this study. Any analysis, findings, conclusions or recommendations expressed in this paper are those of the authors alone and do not necessarily reflect the views of the data sponsors.

REFERENCES

- [1] Y. Gu, Z. Qian, and F. Chen, "From Twitter to detector: Real-time traffic incident detection using social media data," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 321–342, 2016.
- [2] B. Anbaroglu, B. Heydecker, and T. Cheng, "Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 47–65, 2014.
- [3] R. Dowling, A. Skabardonis, M. Carroll, and Z. Wang, "Methodology for Measuring Recurrent and Nonrecurrent Traffic Congestion," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1867, pp. p. 60–68, 2004.
- [4] O. Deniz and H. B. Celikoglu, "Overview to some existing incident detection algorithms: a comparative evaluation," *Procedia Social and Behavioral Sciences*, pp. 1–13, 2011.
- [5] T. Lomax and R. Margiotta, *Selecting travel reliability measures*. the Institute, 2003.
- [6] M. Bani Younes and A. Boukerche, "A performance evaluation of an efficient traffic congestion detection protocol (ECODE) for intelligent transportation systems," pp. 317–336, 2015.
- [7] S. Yang, "On feature selection for traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 160–169, 2013.
- [8] X. Zhang, E. Onieva, A. Perallos, E. Osaba, and V. C. Lee, "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 127–142, 2014.
- [9] X. Kong, Z. Xu, G. Shen, J. Wang, Q. Yang, and B. Zhang, "Urban traffic congestion estimation and prediction based on floating car trajectory data," *Future Generation Computer Systems*, vol. 61, pp. 97–107, 2016.
- [10] Y. Sugiyamal, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S. I. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks-experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, 2008.
- [11] A. Sobral, L. Oliveira, L. Schnitman, and F. D. Souza, "Highway Traffic Congestion Classification using Holistic Properties," *Computer Graphics and Imaging / 798: Signal Processing, Pattern Recognition and Applications*, vol. November 2015, pp. 458–465, 2013.
- [12] X. Yu, S. Xiong, Y. He, W. E. Wong, and Y. Zhao, "Research on campus traffic congestion detection using BP neural network and Markov model," *Journal of Information Security and Applications*, vol. 31, pp. 54–60, 2016.
- [13] A. Mohan Rao and K. Ramachandra Rao, "Measuring Urban Traffic Congestion - A Review," *International Journal for Traffic and Transport Engineering*, vol. 2, no. 4, pp. 286–305, 2012.
- [14] N. Isa, M. Yusoff, and A. Mohamed, "A Review on Recent Traffic Congestion Relief Approaches," in *Proceedings - 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, ICAIET 2014*, 2015, pp. 121–126.
- [15] N. Mazzenga and M. Demetsky, "Mitigation of Recurring Congestion on Freeways," in *Transportation Research Board 89th Annual Meeting*, vol. 434, 2010, p. 16.
- [16] M. Yildirimoglu and N. Geroliminis, "Experienced travel time prediction for congested freeways," *Transportation Research Part B: Methodological*, vol. 53, pp. 45–63, 2013.
- [17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [18] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS ONE*, vol. 10, no. 3, p. e0119044, 2015.
- [19] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," *Proceedings of the IEEE International Conference on Data Mining, ICDM*, pp. 499–508, 2017.
- [20] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.
- [21] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [22] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems 25 (NIPS2012)*, pp. 1097–1105, 2012.
- [23] L. Deng, "Deep learning: from speech recognition to language and multimodal processing," *APSIPA Transactions on Signal and Information Processing*, vol. 5, 2016.
- [24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.
- [25] J. Han, R. Krishnan, J. W. Polak, and J. Barria, "A New Method for Probabilistic Traffic State Identification Using Loop Detector Data: Theory and Empirical Results," in *UTSG*, Plymouth, UK, 2010.
- [26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, no. 3, pp. 807–814, 2010.
- [27] Tieleman, Tijmen, and G. Hinton, "Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude," in *Coursera: Neural networks for machine learning*, 2012, pp. 26–31.
- [28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ImageNet Challenge*, pp. 1–10, 2014.
- [29] J. Lu, S. Chen, W. Wang, and H. Van Zuylen, "A hybrid model of partial least squares and neural network for traffic incident detection," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4775–4784, 2012.
- [30] Q. Liu, J. Lu, and S. Chen, "Traffic incident detection using random forest," in *Transportation Research Board 92nd Annual Meeting*, Washington, DC, 2013, pp. 13–1610.