DOI: xxxx/xxxx

ORIGINAL ARTICLE

Title: A Feature Extraction and Deep Learning Approach for Network Traffic Volume Prediction Considering Detector Reliability

Xiexin Zou[1] | Edward Chung[1] | Yue Zhou[1] | Meng Long[1] | William H.K Lam[1]

[1]Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong, China

Correspondence
Edward Chung, Department of Electrical Engineering, The Hong Kong Polytechnic University Email: edward.cs.chung@polyu.edu.hk

ABSTRACT

Accurate traffic volume prediction plays a crucial role in urban traffic control by relieving congestion through improved regulation of traffic volume. Network-level traffic volume prediction and detector failure have rarely been considered in the literature. This paper proposes a framework based on long short-term memory and the multilayer perceptron that can predict network-level traffic volumes even with detector failure. A profile model learns the profile of the detector's signature (traffic pattern). Detectors with similar profiles are considered to have similar traffic patterns and are grouped into a cluster. Failed detectors can obtain reference information from similar detectors in the same cluster without additional information. A predictive model is developed for each cluster. The proposed method is validated using Japan Road Traffic Information Center data for three cities. The computational results indicate that the proposed method performs well both on typical days and atypical days (COVID-19 lockdown period and the 2021 Tokyo Olympics). Further, it considers detector reliability: the increase in MAE is less than 1 veh/ 5 min when the probability of detector failure increases to 20%.

1 INTRODUCTION

Traffic congestion has become a serious problem for urban traffic systems, one that affects the safety, economy, and environment of cities (Albalate & Fageda, 2019; Fernandes et al., 2017). Traffic signal control at intersections is a low-cost means of regulating traffic flow and alleviating traffic congestion. In the era of big data, the real-time control method is gradually changing to proactive control. The possible interference or congestion in the near future is predicted through the detection and analysis of the current traffic. Measures can be taken in advance to improve the real-time control and service level. In addition, public transportation is gradually becoming demand-responsive. It is increasingly required to accurately estimate transport demand in advance to maximize resource utilization and minimize the waiting time (Peled et al., 2021). Proactive traffic signal control algorithms are especially effective as they can better adapt to the time-varying characteristics of road traffic flow by using anticipated traffic volumes (Wang et al., 2019; Zheng & Liu, 2017). With the maturity of data collection technology, data-driven methods have become increasingly popular mainstream traffic volume prediction methods because of their ability to identify traffic characteristics buried in historical data. Traffic volume data are typically collected by inductive loop detectors deployed on road networks. However, these detectors can fail (Treiber et al., 2011; Turner et al., 2000). Although many advanced data-driven traffic volume prediction models have been developed, when the detector fails, no data are measured and therefore the model receives no input. Moreover, the current traffic volume prediction methods mainly target a single detector, or at most a single road section. In light of the above, this study aims to develop a deep learning (DL)-based method for network-level traffic volume prediction that exploits the rich temporal and spatial characteristics buried in historical data and provides reliable predictions for the whole network even when detectors fail.

The remainder of this paper is organized as follows. Section 2 reviews relevant studies. Section 3 describes the

data and the data processing method. Section 4 presents the proposed DL-based network traffic volume prediction method. Section 5 describes the experiments, and Section 6 presents the conclusions.

2 RELATED WORKS AND CHALLENGES

2.1 Capturing Temporal Relationships

Traffic volume data consist of a set of time series. Although the traffic conditions at each location constantly change, the traffic state of each time slice is closely related to its most recent past state. This temporal relationship is usually captured using either parametric or nonparametric methods. The conventional parametric methods used to characterize the temporal relationships of a single detector are the auto-regressive integrated moving average model and exponential smoothing (Williams et al., 1998). These methods perform well under smooth traffic conditions. They can also analyze and identify traffic flow noise and singularities. For example, Jiang & Adeli (2004) used wavelet packets and autocorrelation functions to analyze traffic flow time series. However, they lack responsive predictive capability, as they cannot handle non-recurrent events.

Nonparametric methods can be divided into conventional machine learning methods and artificial neural networks. Conventional machine learning methods, such as support vector machines, perform poorly because of the highly nonlinear characteristics of traffic data. They also struggle with multidimensional data. Hence, conventional machine learning methods are not popular in the current era of big data.

To better capture the highly nonlinear characteristics of traffic data, researchers have used various types of neural networks, such as the convolutional neural network (CNN), recursive neural network (RNN), and long short-term memory (LSTM). Many studies have confirmed that neural networks can capture the unstable and random nonlinear changes in traffic data better than conventional methods (Kuang et al., 2010; Lefèvre et al., 2014; Vlahogianni et al., 2004).

In addition to traffic prediction, DL methods are applied in various tasks, such as security (Bui et al., 2022; Xu et al., 2021), traffic incident detection (Samant & Adeli, 2000), and track irregularities inspection (C. Li et al., 2022).

Lv et al. (2015) applied the autoencoder, a DL method, for freeway traffic flow prediction. Their results confirm that the method identified potential traffic flow feature representations, such as nonlinear spatiotemporal correlation, from traffic data. S. Zhang et al. (2020) constructed a 3D-CNN model and used a three-dimensional convolution kernel to extract spatiotemporal relationships for urban expressway traffic speed forecasting. The outcomes indicate that CNN can capture local relationships between neighbors and learn the temporal relationships between adjacent time slices. These findings are consistent with the advantages of CNN researchers have found in other applications (Lee et al., 2018; Li et al., 2020; J. Zhang et al., 2020). He et al. (2019); Pang et al. (2019) applied RNN to extract features in the time dimension for bus travel time and arrival time prediction. The results reveal that RNN's strength lies in learning time series-related tasks. However, RNN suffers from the vanishing gradient problem, which effectively terminates the learning process. LSTM was proposed to solve this problem. LSTM-based models, similar to RNN, process sequential data. Currently, LSTM is the most widely used temporal relationship extraction method. Abduljabbar & Dia (2021) developed a speed prediction model for specific location experiments and confirmed the excellent performance of the LSTM model in capturing temporal and spatial traffic dynamics. Abduljabbar et al. (2021) applied LSTM to predict the traffic speed and flow of a single road and evaluated it on the Pacific Motorway between Brisbane and the Gold Coast in Queensland, the Tullamarine Freeway in Melbourne, and the South Eastern Freeway in Melbourne. Their results show that LSTM and its variants effectively capture short-term changes in traffic data.

DL technology is currently the most popular method for learning temporal relationships. Among them, LSTM has an overwhelming advantage in various short-term prediction tasks. In addition to the historical traffic state, data from neighboring detectors can be used to improve the prediction. Hence, correlation among similar detectors needs to be considered in traffic volume prediction. When a detector fails, the model can obtain reference information from correlated detectors. The literature generally considers this correlation between detectors to be a spatial relationship.

2.2 Capturing Spatial Relationships

A large city has multiple heterogeneous activity centers. However, observations from detectors adjacent to the same type of activity center may have a homogeneous travel pattern. To capture this homogeneity (i.e., the spatial relationships among detectors) and improve the accuracy of traffic predictions, traffic data from multiple detectors are often presented as grid-based or graph-based input data.

Grid-based traffic data are expressed as a regular image, with a pixel representing a data record. CNN can thus obtain the local spatial relationships between adjacent pixels. For example, Zang et al. (2019) constructed a spatiotemporal matrix to represent historical traffic speed data. The rows represented the locations of detectors on the road, and the columns indicated time. There is also much work to represent the traffic data as a 3-dimensional tensor. Each channel represents the traffic information of all grids on a time slice (S. Guo et al, 2019; Liu et al, 2021; S. Zhang et al, 2020). CNN and RNN were then integrated to predict traffic speed for a single road. Yao et al. (2019); Zheng et al. (2019) applied CNN and LSTM to forecast grid-based traffic flow. The results indicate that the combination of CNN and RNN/LSTM effectively obtains spatiotemporal relationships for grid-based prediction. In addition, some researchers have

TABLE 1. Related work on network-scale traffic prediction.

Work	Model	Output/ Task	Limitation
Cui et al. (2020)	LSTM	Volume prediction of multiple detectors on four connected highways.	Input/Output that requires regular representation. Arrange detectors according to the sensor location.
Han et al. (2019)	CNN	Traffic volume prediction of a single detector.	Space relationship is not considered.
Tarunesh & Chung (2020)	AE, LSTM	Traffic volume prediction of strongly associated detectors in small-scale net-	Applicable to small-scale networks with strongly associated detectors
Liu et al. (2021)	CNN	Traffic demand forecast for each grid (subregion).	Need to compress data resolution, input/output of regular representation.
Deng et al. (2022); Grötschla & Mathys (2022); J. Li et al. (2022); X. Wu et al. (2022)	GNN	Travel time of road segments between important urban intersections.	Learning large-scale graphs is challenging.
Proposed	MLP, LSTM	Traffic volume predictions for all detectors.	

combined other DL modules to strengthen predictions. Zhou et al. (2022) grid the urban traffic flow into a three-dimensional spatiotemporal tensor. The attention mechanism, CNN-based ResNet, and LSTM are combined to capture each city area's inflow/outflow changes. The aforementioned studies assume that the changes in traffic demand between geographically adjacent areas are similar, limiting the method's applicability to traffic flow predictions between regions. CNNs require the input to be a regular matrix or tensor. It is challenging to arrange all the detectors on the urban traffic network into a regular matrix form. Also, this does not match the real topology of the traffic network. For example, two adjacent detectors in different directions may have significant differences in traffic patterns despite being close in Euclidean distance.

The irregularity of the transportation network topology makes it very difficult to manually construct network-level regular inputs. To satisfy this requirement, graph neural networks (GNNs) were introduced to predict traffic flow and speed from graph-based data (Diao et al., 2019; Fang et al., 2019; Yu et al., 2018). With graph-based data, each investigated location is treated as a node, and the spatial relationships between adjacent nodes are captured. With graph convolutional networks (GCNs), spectrogram theory can be used to extend the convolution operation to the non-Euclidean domain. However, if more GCN layers are added, data will be overfitted. Moreover, the number of neighbors of each node in GCN is fixed, which does not reflect the actual traffic conditions. GCN makes full use of the weight information between edges. In addition to GCN, graph attention networks (GAT) use an attention mechanism to implement feature analysis and aggregates the features of neighbors to construct the feature of the target node (Do et al., 2019; G. Li et al., 2021; Q. Song et al., 2020). Different from GCN, the graph convolution kernel of GAT is dynamically obtained according to node features. However, GAT uses the connectivity between nodes and ignores the edge information. For GNNs, the construction of the graph is crucial. There needs to be an edge connection between each node and their similar nodes while avoiding too many redundant edges. Therefore, sufficient prior knowledge is required to avoid constructing suboptimal graphs. Obtaining additional data to build is time-consuming and costly for all detectors in an urban traffic network. The calculation of a huge graph is also very challenging.

2.3 Network-Level Prediction

In addition to extracting spatiotemporal correlations from traffic data for accurate prediction, researchers have focused on network-level traffic prediction. Many studies have focused on highway data, but urban traffic flow, which is more complex because of traffic signal control, is rarely studied. Vlahogianni et al. (2014) reported that the interactions in densely populated urban road networks make network-level prediction challenging.

Thousands of detectors are installed in large cities, and thousands of corresponding models are developed. However, a model will be unable to provide predictions if a detector fails. The popular approach to network-level traffic volume prediction is to train N models for N detectors (Duan et al., 2016). Table 1 lists research on network-level traffic prediction and the limitations that make them unsuitable for the prediction task in this paper.

Cui et al. (2020) represented the speed data of 323 detectors on four connected freeways as a vector of sensing locations and used LSTM to achieve good speed prediction. The authors concluded that the locations of detectors have little effect on the speed prediction results. Tests conducted on the INRIX dataset (which includes both freeway and urban roadway segments) led to the same conclusion. However, the authors stated that the relatively small variation in the INRIX speed data could explain their conclusion,

which therefore would not apply to network-level traffic volume forecasts as the traffic volume of urban roads has high variance due to traffic control.

Han et al. (2019) proposed DeepCluster to reduce the number of models required for network-level traffic volume prediction. They studied a transportation network consisting of 27 roads. Specifically, they constructed triplets of visual images generated from the daily time series of each detector. Each triplet contains two images of the same detector and one image of the other detector. Then, CNN and triplet loss were used to extract features from each daily image. Each detector is represented by the average of the features obtained from all its images. K-means clustering was applied to cluster all the detectors into three groups. The detectors in each cluster share a single-detector predictive model trained on data from detectors on all roads in that cluster.

Tarunesh & Chung (2020) used neural networks to model the traffic network and autoencoders to reduce the model size by exploiting the spatial correlation between detectors. The author removed some negatively correlated detectors by observing the correlation coefficient of the detectors' time series. Hence, there is a strong correlation between most of the studied detectors. Finally, AE and ANN are used for prediction work. This work is suitable for small-scale networks where detectors are strongly correlated.

Due to the challenges posed by the calculation of large-scale graphs, Liu et al. (2021) weighed the data resolution and task complexity and divided the data of a time slice of the whole city into a grid representation by analyzing and removing some detectors. The input is defined as 3-channel data. Each channel represents a certain type of data (speed/flow/direction) of a certain historical time slice. The U-Net-based structural design framework is used to predict the traffic state.

Unlike small subsystems, which consist of only a few road segments, there are currently competitions dedicated to exploring the dynamics of traffic states across the entire city. Currently, traffic volume and travel time prediction are usually performed independently. The Traffic4cast 2022 competition provides world data from industrial-sized fleets and vehicle counters, allowing participants to predict congestion classes (red/yellow/green) and supersegment-level travel time.

X. Wu et al. (2022) applied multiple KNN filters to obtain features based on traffic state similarity. Then combined them with static features such as speed limits and flow between intersections and built a non-deep model, a gradient-boosting tree model for prediction. Deng et al. (2022) treat intersections as nodes and embed multiple features for nodes. For example, historical one-hour traffic data, time information, speed_kph, highway and oneway, etc. GAT is introduced to fuse the features between nodes and strengthen the correlation between learned representations. However, significant overfitting is observed in some cities. J. Li et al. (2022) regard road segments as nodes and use GNN to obtain the spatial relationship between road segments and make predictions. In addition, the authors also cluster all records into multiple equal-frequency bins by sorting the volume sum of all detectors. This enables each node to contain congestion information under different global traffic volumes, effectively improving the model's performance. Grötschla & Mathys (2022) compressed the road graph to simplify the flow information and applied a GNN-based model and a multi-task method to predict the congestion level and ETA simultaneously.

The above research explores the travel time of road sections between important intersections in the city. However, it is still difficult to extract the traffic volume changes of all detectors in the network. The travel behavior of the road network is complex and dynamic, and it is difficult to capture all the features of all detectors with one graph. Using all detectors as nodes will result in a very large graph. Moreover, learning such a huge graph is also very challenging and easy to overfit. By mining as much information behind all detectors as possible, we can better grasp the evolution of urban traffic patterns.

2.4 The Proposed Method: Challenges and Contributions

The foregoing sections have described the challenges facing traffic volume and speed prediction. Various methods have been developed to capture the spatiotemporal relationships between traffic data, with varying success. Most predictive models are for a single intersection or corridor, but some models can predict the whole network in real time. The following are some key unresolved challenges:

(1) Utilization of the full potential of artificial intelligence

Although LSTM has proved very effective for time series-related problems, appropriate low-dimensional input must be defined to ensure accuracy in real-time prediction. Converting the historical traffic volume of the entire network into a suitable input is the challenge that must be overcome to use LSTM for prediction at the network level.

(2) Responsive prediction

Most studies do not consider the model's responsiveness to atypical cases, such as COVID-19 lockdowns and the 2021 Tokyo Olympics. Data-driven models capable of capturing latent associations between detectors in transportation networks may generate more responsive predictions.

(3) Detector failure

Most current studies test predictive models using complete input and evaluate their performance using metrics such as the mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). They do not analyze the robustness of the model against missing input, e.g., due to detector failure.

(4) Large-scale network traffic volume prediction

Capturing the spatial relationship of all detectors in the entire traffic network to aid in volume prediction is challenging. First, detectors are difficult to arrange into a regular matrix representation without compressing the resolution. If real traffic graphs are used directly, the position

closeness cannot correctly represent the similarity of the traffic pattern of detectors (Section 2.2). Moreover, the calculation of large-scale graphs is still a challenge. Second, predicting volumes for all detectors using a single model requires embedding spatial relationships in the input, which typically need to be extracted from other sources. Moreover, the model will be complex and delicate, which is also a challenge for design and learning.

In response to the above challenges, this paper proposes a framework based on LSTM and the multilayer perceptron (MLP) that can quantitatively analyze and predict network-level traffic volumes even when some detectors fail. Specifically, to obtain the latent associations between detectors in a transportation network, a profile model is trained to recognize detectors' profiles. The shallow layers of the trained profile model are used to extract all the detectors' features. Hierarchical clustering is then applied to build multiple clusters based on their features and finally develop an LSTM-based model for each cluster. Each cluster consists of detectors with similar profiles, i.e. strong spatial correlation. Each predictive model corresponds to certain homogeneous detector profiles.

To evaluate the proposed method, we use the trained model to generate predictions on typical days and atypical days (COVID-19 lockdown and the 2021 Tokyo Olympics). In particular, we evaluate the model's performance when detectors fail.

The proposed method is applied to Japanese cities of various sizes-Tokyo, Osaka, and Kochi-to test its effectiveness and robustness. The traffic volume data used are sourced from the Japan Road Traffic Information Center.

3 Problem and Data

3.1 Problem Statement

Problem: traffic volume prediction. Given k (k=6) h historical volumes of each detector, to predict the volumes of all detectors in the following time step (time interval is 5 minutes). X_t is an N-dimensional vector representing volumes of N detectors at time slice t.

A function f(·) is expected to be learned:

$$f : [X_{t_k}, , X_{t-2}, X_{t-1}] \Longrightarrow X_t \quad (1)$$

3.2 Data

The three cities studied-Tokyo Metropolis and Osaka and Kochi prefectures-have varying population sizes, population density, and number of traffic detectors. Tokyo Metropolis has 14 million residents, while Osaka and Kochi prefectures have 2.7 million and 758,000 residents, respectively. Data used to develop the model were collected at 5 min intervals in 2020; 80% of the data is used for training and 20% for validation. Data from the COVID-19 lockdown period (January to May) and the Tokyo Olympics (July 22 to 24) in 2021 are used for testing.

Missing data are unavoidable as communications interruption, detector failure, or detector malfunction are inevitable. Therefore, data must be cleaned and completed before being used for training and testing. Fig. 1 shows the working status of the detectors in Tokyo. There is no detectors with 0% failure and 82% of the detectors have less than 20% failure. For the remaining 18%, most are installed on the outskirt and often fail during peak hours. Accordingly, detectors with failure rates higher than 20% are removed. Our input data for Tokyo, Osaka, and Kochi DL models are supplied by 1,888, 1,831, and 256 valid detectors, respectively.

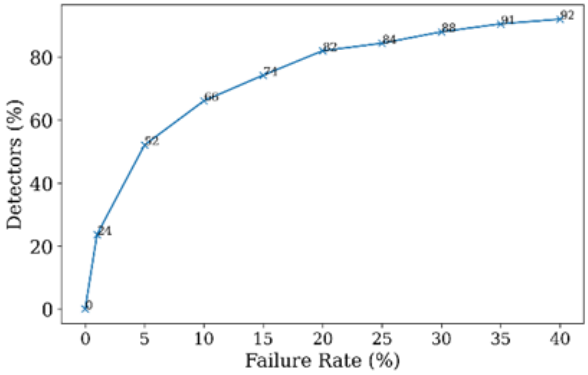


FIGURE 1. Failure of the Tokyo detectors

After removing invalid detectors, 2% to 5% of the detectors still fail randomly in each time slice. The failure durations of the detectors range from a few minutes to a few hours. Training the DL model using these incomplete raw data will result in errors. Hence, the missing data are filled using the k-nearest neighbors (KNN) algorithm.

Data of a failed detector can be filled according to the volume of its upstream or downstream detectors on the same lane. However, in this study, the specific positions of some detectors are unknown. Moreover, in most cases, neighbor detectors also have failures. Therefore, the dataset is divided into multiple subsets according to the date. Each subset contains the data of all detectors on the same day. For most cases where the detectors fail only part of the day, KNN is used for filling. For each day, KNN finds the most similar k detectors for each failed detector, then uses the mean value of these k detectors to fill. In this work, k was set to 3. Specifically, the Euclidean distance between each volume reported by the failed detector on the day and the volumes of other detectors at the corresponding time is calculated. The distance between the failed detector and others on the day is the mean value of the obtained corresponding distances. The smaller the distance, the more similar the two detectors are on the day. However, the above KNN-based methods cannot fill all null values. If a detector is not working for an entire day, KNN cannot calculate its distance from other detectors. In another case where all detectors fail in a time slice, no measurements can be used to fill in that time slice. In both cases, the imputation is performed using observations from the past 15 minutes. If the past 15 minutes observation are not available, similar day of the week and type of day (holiday, non-holiday) of most recent week is used.

When performing model testing in Section 5, only observed data are considered.

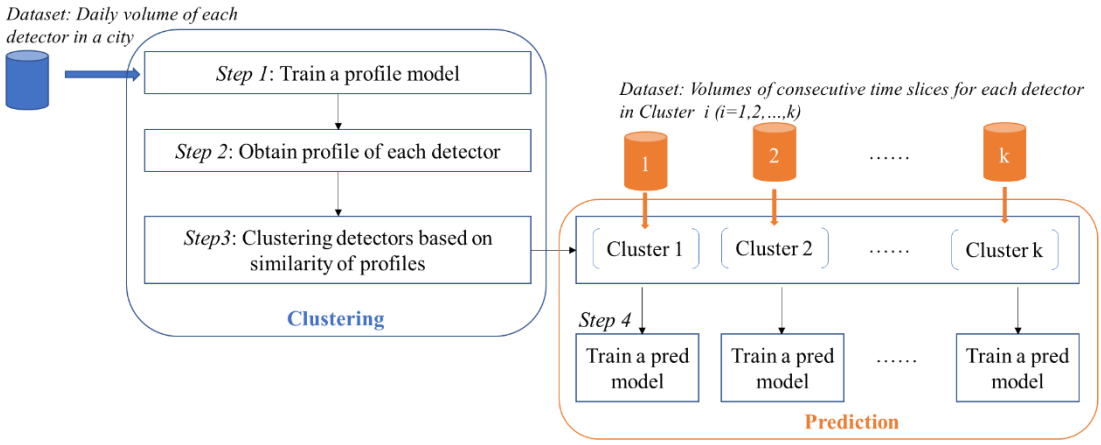


FIGURE 2. Overall flow

4 Methodology

As shown in Fig. 2, the research methodology comprises two stages. First, detectors with homogeneous features are extracted and clustered for faster and better learning and for reducing the model's input. Second, a DL model to predict the traffic volume is developed for each cluster.

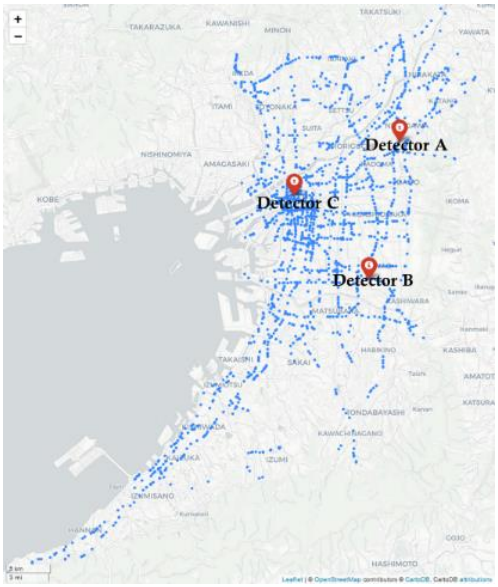


FIGURE 3. Map of Osaka, with the three selected detectors marked

4.1 Clustering Method

The road network of large cities has over a thousand detectors; e.g., Osaka has 1,831 detectors, as shown in Fig. 3. Due to the complex travel behavior of the residents, a variety of traffic patterns are recorded in all detectors. For example, the varying travel patterns of three of Osaka's detectors (A, B, and C), whose locations are shown in Fig. 3, are depicted in Fig. 4.

Therefore, using only one model to extract all the different traffic patterns will result in the characteristics of a prevalent traffic pattern being learned while those of other patterns are ignored. Moreover, with high-dimensional input data that contain redundant information, learning is challenging, and therefore the resulting DL model will be more complex. The advantages of DL cannot be fully utilized if only one DL model is used for the whole network. High-dimensional input and complex DL models will also increase the prediction time, which will handicap the real-time application of the model. If a separate model is built for each detector, thousands of models will be required in a network. Moreover, the model will not be able to provide valid predictions when a detector fails as no reference data can be used as input. As shown in Fig. 4, although some detectors (A and B) are not geographically adjacent in the urban road network, they show similar traffic trends; we assume this is because they share some homogeneous features. When detector A fails, detector B can be referenced.

For network-scale prediction, it is necessary to minimize the number of models that need to be developed while ensuring accuracy. To achieve this goal, we propose clustering detectors with homogeneous features together for mutual reference and isolating heterogeneous detectors to prevent noise. Therefore, it is crucial to capture the features of the detectors well and use them to analyze the relationships between the detectors. This study utilizes a DL model to extract each detector's daily data features, which are concatenated to construct the detector's characteristic (see Fig. 5). Homogeneous detectors are clustered using hierarchical clustering based on the characteristic distance. Details of the feature extraction, distance calculation, and hierarchical clustering are given below.

4.1.1 Feature Extraction

Each detector has some inherent profiles/characteristics. For example, detectors near a commercial center will reflect the ebb and flow of activities in that center, and its characteristics will differ from those of detectors near schools. The homogeneity of detectors' characteristics makes their hidden features similar.

A DL model is applied to learn and extract the detectors' features. A few fully connected (FC) layers models, as shown in Fig. 6, are used for the feature extraction. For N detectors in a dataset, there are $128, 64, 128$, and N neurons in each

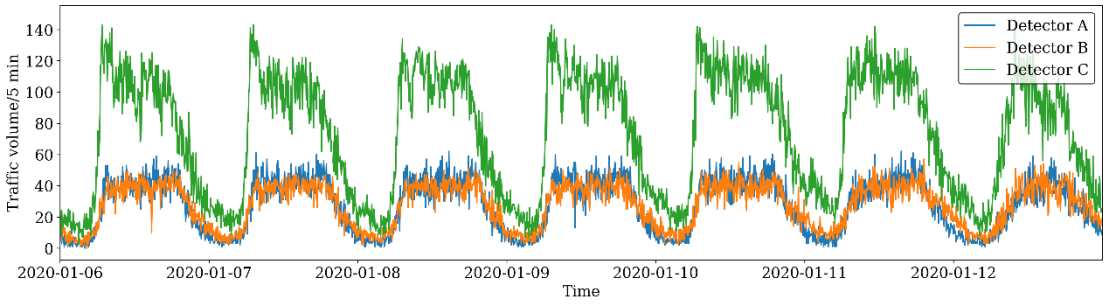


FIGURE 4. Visualization of observed traffic volumes of three detectors for a randomly selected week in Osaka

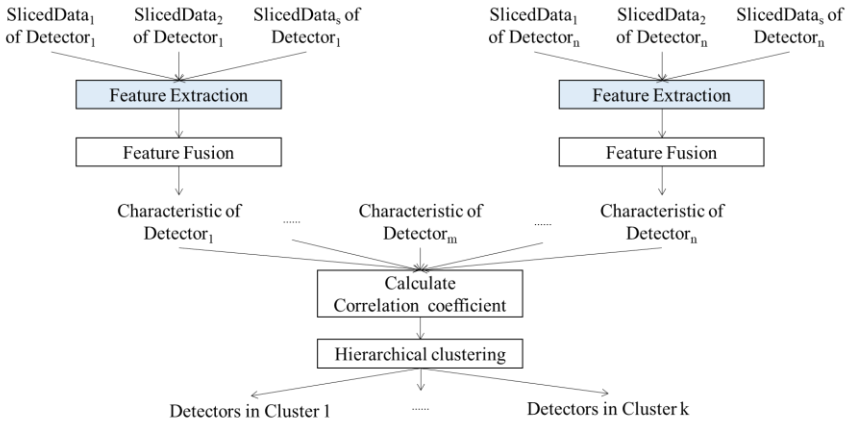


FIGURE 5. Flow of the clustering method

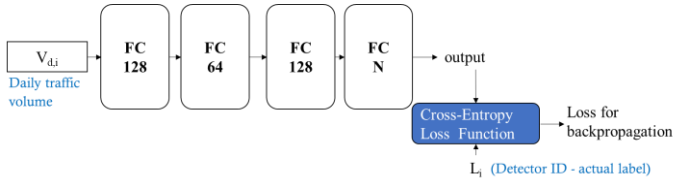


FIGURE 6. Clustering: feature extraction model

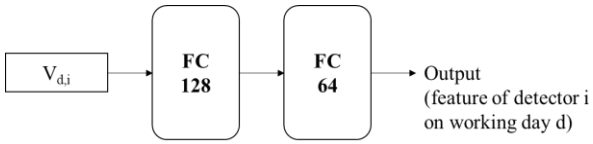


FIGURE 7. Clustering: features extracted from the trained deep learning model

layer, respectively. The goal is to map the daily traffic volume, $v_{d,i}$, from detector *i* on working day *d* to a probability distribution to represent that detector.

The input consists of all the working days in the training set. Each detector records multiple daily traffic volume sequences, and each day is represented by a 288-dimensional vector (i.e., each vector represents 5-minute traffic volume data). The output is the label assigned to the detectors. All inputs are normalized to the range [-1,1]. The outputs of the first three FC layers are sent to the rectified linear unit (ReLU) activation function to become the input of the next layer. As the goal of the network is to determine which detector the input time series originates from, the cross-entropy loss of the multi-classification network (De Boer et al., 2005) is applied.

The shallow layers of the DL model learn more general features, whereas the deeper layers learn more targeted details (Neyshabur et al., 2020; Ribeiro et al., 2016; Yosinski et al., 2014). This model aims to learn the general features that are suitable for clustering. After the model training is completed, the outputs of the middle layer (64 neurons) are selected (as shown in Fig. 7).

In addition to the above profile model, Auto Encoder has also been tried to extract important compressed features from each daily volume. This method aims to compress the input to a smaller size, and the compressed features can also be restored to the input. However, since this method cannot learn the signature of each detector, it is not as good as the classification effect of the above profile model.

4.1.2 Distance Calculation

After feature extraction, each daily traffic volume sequence is compressed into a 64-dimensional vector. The characteristic of each detector is determined by concatenating the features extracted by the above model from all working days. Subsequently, the correlation coefficients of all the detectors' characteristics are calculated using Eq. (2):

$$\rho_{xy} = \frac{cov(x,y)}{\rho_x \rho_y}, \tag{2}$$

where *cov(x,y)* is the covariance between detectors *x* and *y*, and ρ_x and ρ_y represent the standard deviation. When ρ_{xy} is closer to 1, the detectors x and y are more correlated; that is, the distance between *x* and *y*, $distance_{xy}$, is smaller. Therefore, Eq. (3) is used to convert the correlation to a distance:

$$distance_{xy} = exp^{-\rho_{xy}} \tag{3}$$

TABLE 2. Number of detectors in each cluster

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
Tokyo	236	363	193	167	241	75	281	75	65	236	363
Osaka	215	112	329	171	260	114	119	85	192	87	147
Kochi	108	98	50								

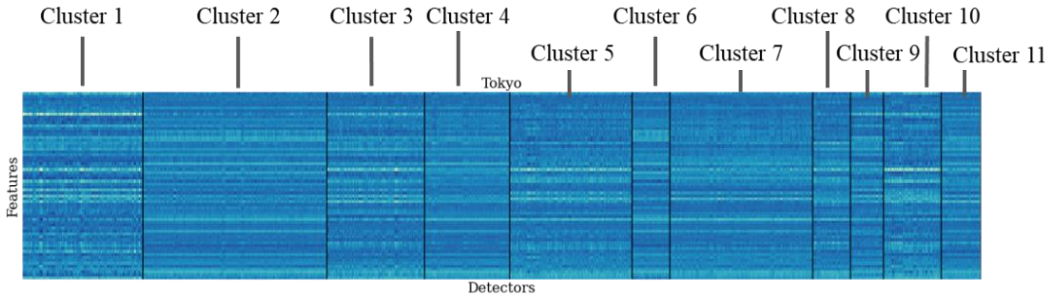


FIGURE 8. Visualization of the features of the detectors in each cluster in Tokyo

4.1.3 Hierarchical Clustering

Hierarchical clustering (Johnson, 1967) is applied to cluster detectors using the obtained distances (Eq. (3)). The hierarchical clustering steps are as follows:

a) Classify each detector into one category, to obtain *N* categories. Each category contains only one detector.

b) Find the two closest clusters and merge them, decreasing the total number of clusters by one.

c) Recalculate the distances between the new clusters and all other clusters.

Repeat steps b) and c) until the specified number of clusters is obtained.

Fig. 8 visualizes the features of all the detectors in Tokyo extracted from a randomly selected working day. The y-axis represents the feature, and the x-axis represents the detector. All the detectors are rearranged cluster-wise, the clusters being separated by black vertical lines. There is greater similarity between the extracted features of detectors in the same cluster than with the extracted features of detectors in other clusters.

A large number of clusters means that the detectors in each cluster's characteristic are more closely related (smaller distance), implying that learning the features of each cluster will be easier and faster. However, a large number of clusters could result in clusters containing only a few detectors, or even a single detector. For example, when the Tokyo detectors are divided into 25 clusters, there is a cluster with only a single detector. A detector failure in a single-detector cluster will result in a 100% failure rate; i.e., no other detector in the cluster can be referenced. Hence, the number of clusters must be traded off against the minimum number of detectors in a cluster. In this study, the minimum number of detectors in each cluster is set to 50. Table 2 lists the number of detectors in each cluster. Section 5 compares the prediction results for different cluster sizes and the minimum number of detectors per cluster.

4.2 Predictive Model

After clustering, each cluster is composed of homogeneous detectors. LSTM is applied to learn their common features from historical data. LSTM is chosen because research (Lv et al., 2018; Yao et al., 2019; Zheng et al., 2019) has shown that it can learn time series well. With high-dimensional input data, the LSTM prediction speed is lower than that of other algorithms (Lee et al., 2018; Li et al., 2020; J. Zhang et al., 2020). However, clustering reduces the dimensions of the input and therefore accelerates the prediction speed, which makes LSTM an even more attractive choice.

The model's architecture is shown in Fig. 9. The inputs, $X_{t-k}(k = 1, 2, ..., 6)$, represent the vector composed of all the detectors in the cluster at time $t - k$. In addition, the hour of day data, a 24-dimensional vector, is included and is denoted by T_t in Fig. 9.

An FC layer is used to process time information to assist in learning the peaks and troughs of the day. After feature concatenation, the second FC layer fuses the multi-source information and obtains the same output size as the number of detectors in the cluster.

The model of each cluster is built according to this structure (Fig. 9). However, each model layer has a different number of units due to the varying number of detectors in each cluster. Each model's input and output dimensions are shown in Table 3, where N_c is the cluster size. $N_c//16$ is the quotient of N_c divided by 16 (see Table 4). Learning and prediction for all clusters can be performed in parallel for the entire traffic network.

.

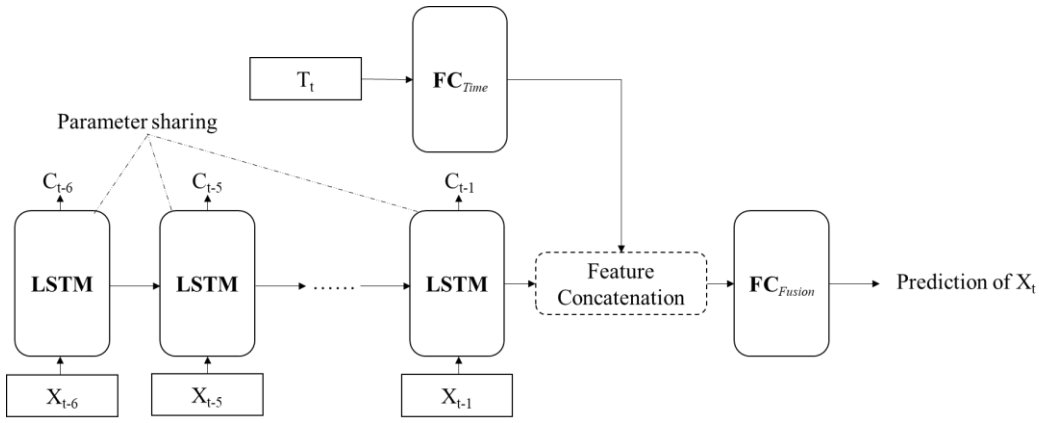


FIGURE 9. The predictive model

TABLE 3. Number of units in each layer

	Input	Output
LSTM	N_c	$(N_c//16+1)\times 16$
FC_{Time}	24	8
FC_{Fusion}	N_c+8	N_c

TABLE 4. Example of $(N_c//16+1)\times 16$ output

N_c Input	Output
$N_c=1$	16
$N_c=16$	32
$N_c=17$	32

5 Experiments

After the model's training is complete, multiple tests are performed to evaluate the accuracy and quality of its prediction. Tests are conducted on Tokyo, Osaka, and Kochi. Three scenarios-one typical (normal) scenario and two atypical scenarios (the COVID-19 lockdown period in Japan) and the Tokyo Olympics days), covering January 11 to 17, 2021, March 29 to April 4, 2021, and July 22 to 24, 2021, respectively-are used.

Min-max normalization is applied to scale the value of the historical traffic volume to the range [-1,1] to increase the gradient descent speed of the search for the optimal solution. The input data are normalized in the test phase, after which the output is re-scaled to the original scale. The metrics are calculated based on the original scale.

The results of the proposed multi-cluster (MC) method are compared with those of the single-cluster (SC) method, which treats all the network detectors as a single cluster. The SC method is similar to the MC method, but clustering is not performed.

All the experiments are conducted on the University Research Facility in Big Data Analytics (UBDA) platform of the Hong Kong Polytechnic University. The CPU and GPU information are as follows: Intel(R) Xeon(R) Gold 6130 CPU @ 2.10 GHz, ASPEED Technology, Inc. ASPEED Graphics Family (rev 41). The DL models are built on the Pytorch framework.

5.1 Metrics

The MAE, RMSE, and MAPE are used to evaluate the predictive model. In addition, Theil's U-statistic (U) (Wheelwright et al., 1984) is calculated to validate the prediction (against the naive method).

$$\mathrm{MAE}(\mathbf{y},\hat{\mathbf{y}}) = \tfrac{1}{N}\textstyle\sum_{i=1}^{N}|\mathbf{y}-\hat{\mathbf{y}}|, \tag{4a}$$

$$\mathrm{RMSE}(\mathbf{y},\hat{\mathbf{y}}) = \sqrt{\tfrac{1}{N}\textstyle\sum_{i=1}^{N}(\mathbf{y}-\hat{\mathbf{y}})^2} \tag{4b}$$

$$\mathrm{MAPE}(\mathbf{y},\hat{\mathbf{y}}) = \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\frac{|\hat{\mathbf{y}}-\mathbf{y}|}{\mathbf{y}}, \{y|y\geqslant 5\}, \tag{4c}$$

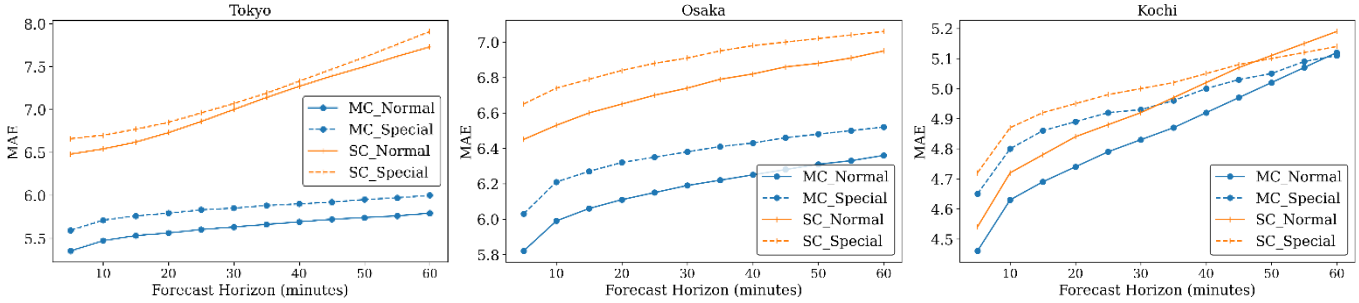
$$\mathrm{U}(\mathbf{y},\hat{\mathbf{y}}) = \sqrt{\frac{\sum_{i=1}^{N-1}(\frac{\hat{\mathbf{y}}_{i+1}-\mathbf{y}_{i+1}}{\mathbf{y_i}})^2}{\sum_{i=1}^{N-1}(\frac{\mathbf{y}_{i+1}-\mathbf{y}_i}{\mathbf{y_i}})^2}} \tag{4d}$$

$\hat{\mathbf{y}}$ and $\mathbf{y}$ represent the predicted traffic volume and the observation, respectively. RMSE is more sensitive to relatively large errors. All calculations of MAPE consider only observations with a traffic volume greater than 5 veh/5 min. This is to avoid division by zero and biases in error for low observed traffic volumes. For example, if the observed traffic volume is 4 vehicles and the predicted traffic volume is 2 vehicles, the MAPE is 50% although the error is only 2 vehicles.

5.2 Predictive Performance and Robustness

The first test assesses the predictive accuracy for time horizons ranging from 5 minutes to 1 hour. When the forecast is conducted for the next 30 minutes and beyond, all input traffic volumes are from the outputs of the previous steps. That is, there are no actual traffic observations. The model's applicability to normal and atypical (special) day scenarios is also tested. Special days test the predictive model's responsiveness and the quality of the features it captures.

The naive method is used as a baseline for comparison, which uses the traffic volume of the current time slice as the prediction of the next time slice. For Tokyo, Osaka, and Kochi, the MAEs of the naive method during normal and atypical (special) days are 7.2/7.5 vehs, 8.5/8.9 vehs, and 6.2/6.3 vehs. RMSEs are 10.37/10.75 vehs, 12.68/13.22 vehs



(a) Tokyo (b) Osaka (c) Kochi

FIGURE 10. Mean absolute error (veh/5 min) results of multi-step prediction (without failures)

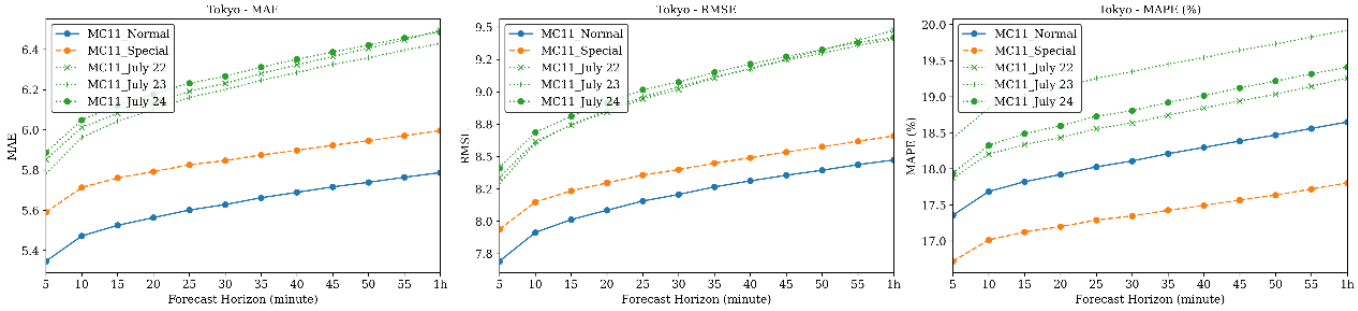


FIGURE 11. Overall prediction results of the multi-cluster method for the 2021 Tokyo Olympics (July 22–24)

and 9.05/9.20 vehs.

For the three cities, the MAEs of the proposed MC method are reduced by 26%-32% compared to the naive method. Specifically, the MAEs of the MC method for Tokyo, Osaka, and Kochi are 5.35/5.59 vehs, 5.82/6.03 vehs, and 4.46/4.65 vehs. The RMSEs are 10.37/10.75 vehs, 12.68/13.22 vehs, and 9.05/9.20 vehs. MAPE is 16.3%/17.2%, 17%/17.2%, 17.5%/18.1%. Fig. 10 shows the MAE obtained by the proposed SC method and the MC method. It shows that the MC method has better predictive performance than the SC method on large cities. As the time horizon increases, the MC method gives better multi-step prediction results than the SC method. At one-hour time horizon, the difference between MAPE results of the SC and MC methods is 5%. For the small city of Kochi, the predictive performance of the MC and SC methods are similar.

In addition to MAE, RMSE, and MAPE, Theil's U-statistic (U) is also applied to measure the predictive accuracy of the proposed method. If U is equal to 1, the predictive performance of the proposed model is equivalent to that of the naive method. The smaller U is, the better the proposed model's predictive performance. Table 5 shows that both the SC and MC methods outperform the naive method, and the MC method achieves the lowest U. When the time horizon is 5 minutes, the naive method performs well, but the prediction becomes unreliable as the time horizon increases to 1 hour.

From the MAE/RMSE/MAPE and Theil's U-statistic, all the DL methods outperform the naive method. And the MC method outperforms the SC method in all three cities. However, the multi-step prediction advantage in a small city is not obvious.

TABLE 5. Theil's U-statistic

Date	Method	Tokyo	Osaka	Kochi
Normal	SC	0.84	0.68	0.70
-	MC	0.68	0.64	0.69
Lockdown	SC	0.81	0.67	0.67
-	MC	0.68	0.64	0.66

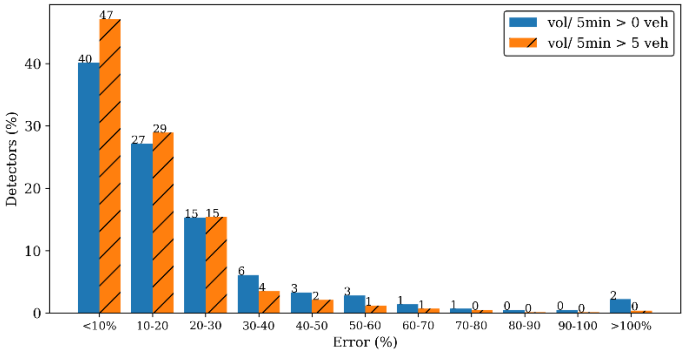


FIGURE 12. Error distribution of MC predictions 2021 Tokyo Olympics (July 22–24)

Tokyo hosted the Olympics in 2021, and the traffic trends during this period differ from the normal trends. The trained models are applied to test the prediction results of the day before and of the first and second days of the Olympics (July 22-24), and the results are shown in Fig. 11. The predictive performance during the Olympics (July 22-24) is slightly worse than on normal days and during the COVID-19 lockdown period in Japan. However, compared with the

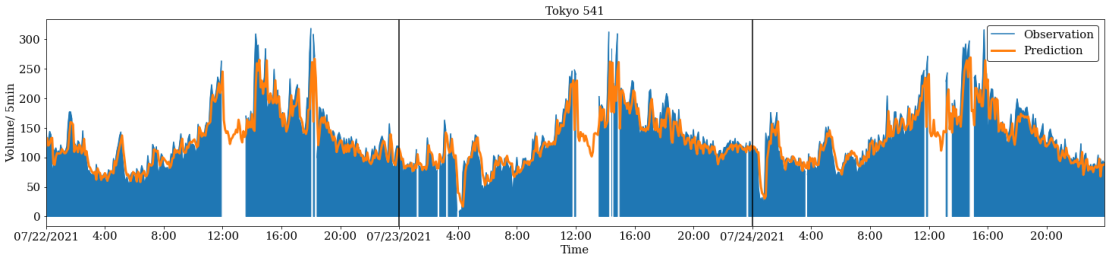


FIGURE 13. Visualization of the predicted traffic volume for the 2021 Tokyo Olympics (July 22–24)

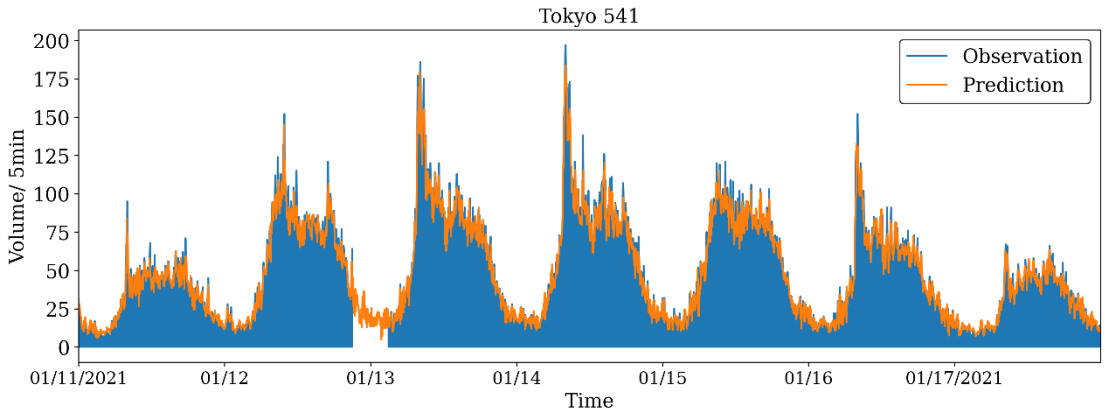


FIGURE 14. Visualization of the predicted traffic volume for normal days (January 11–17, 2021)

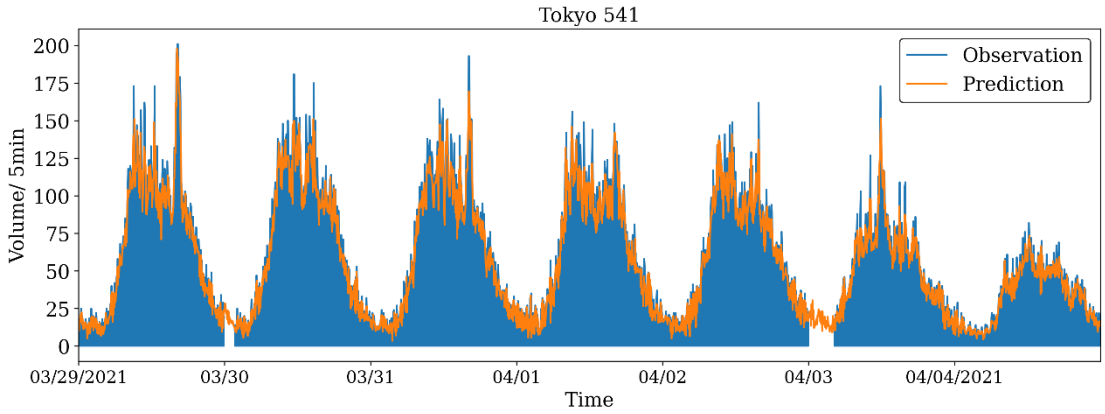


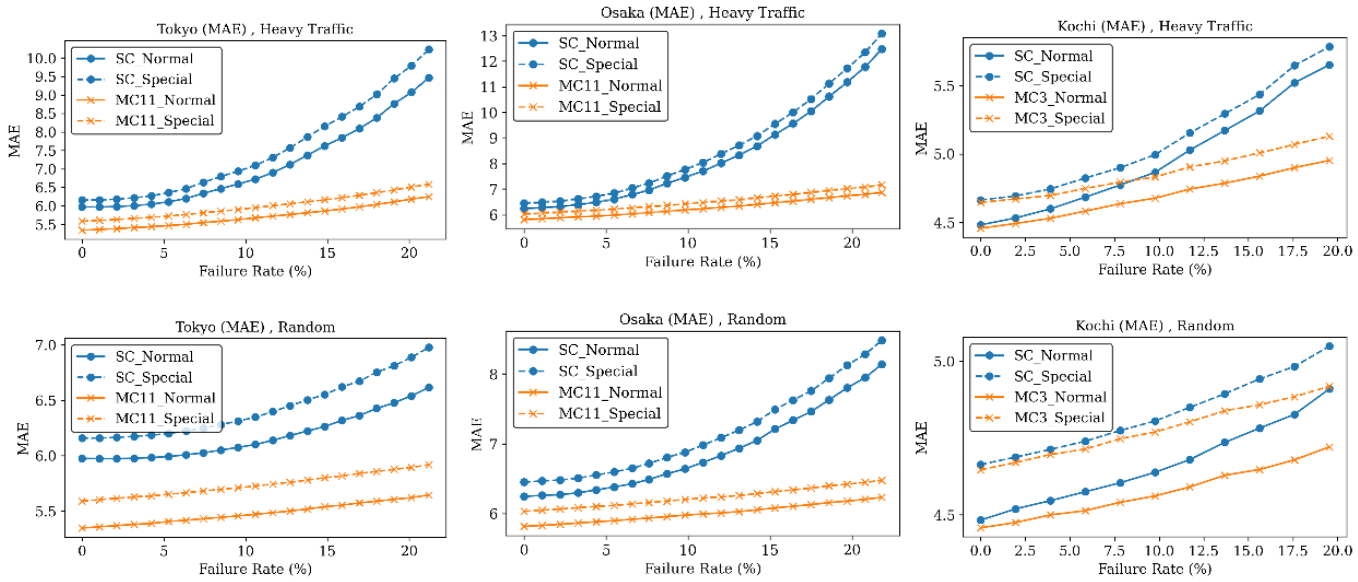
FIGURE 15. Visualization of the predicted traffic volume for lockdown (March 29 – April 4, 2021)

RMSE and MAPE results on normal days, the results are less than 1 veh/5 min and 2% higher, respectively. Fig. 12 shows the error distribution of MC predictions. It can be seen that 67% of the predictions deviate from the observed values by less than 20%. Suppose the case where the actual observation is less than 5 vehs/ 5 min are excluded. In that case, the proportion of prediction error less than 20% improves to 76%.

Fig. 13, Fig. 14 and Fig. 15 show MC predictions of a heavy-traffic sensor in Tokyo during different periods: the Olympics, normal days, and lockdown. It can be seen that different traffic patterns are displayed during different periods. MC methods provide predictions that are close to actual observations for the 3 cases.

Overall, the proposed MC method performs well in different periods, namely normal days, the COVID-19 lockdown period, and the Olympics. Moreover, it is more applicable to large cities. For a small city (Kochi), a single DL model (the SC method) is sufficient to provide reasonable predictions; the MC method produces only a slight performance improvement. Because small cities have fewer types of heterogeneous traffic patterns, a well-designed model can accomplish this task. However, complex and diverse patterns in large cities require a more elaborate predictive model to learn many parameters to capture, which is challenging to design. The MC method uses the profile model and clustering method to identify homogeneous detectors to reduce the complexity of the tasks faced by each predictive model. Each model can learn each similar patterns better to improve the prediction performance.

Furthermore, the proposed method can rapidly predict the whole network for real-time application. The MC method needs 2-4 milliseconds to predict the volume of all the investigated detectors in a city at the next time slice; compared with the SC's 16-20 milliseconds, the MC speed increased by about 80%.



(a) Tokyo (b) Osaka (c) Kochi

FIGURE 16. Comparison of predictions with the multi-cluster and single-cluster methods (with failures)

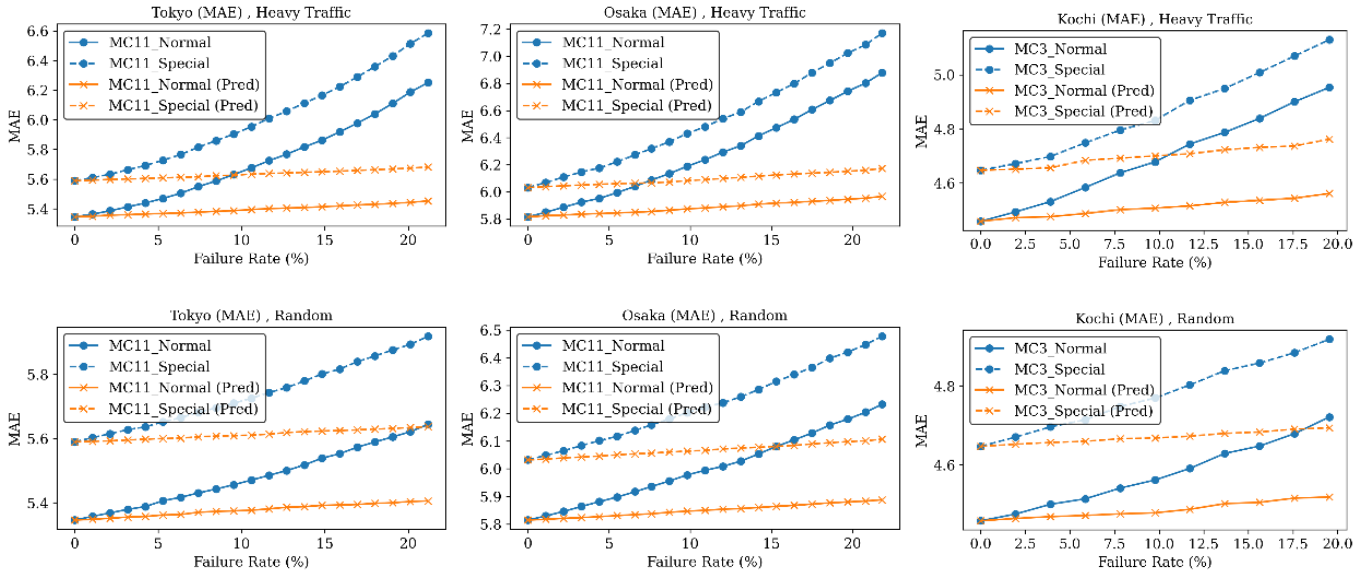


FIGURE 17. Results of the multi-cluster method with different methods of filling missing data

5.3 Failures

Besides reducing the complexity of the prediction task to provide reliable and fast predictions of all detectors in large cities, the novelty of the MC method lies in its resilience to failures. The MC method can provide reliable volume prediction for a detector even if it fails. Random detector failure and heavy-traffic detector failure are tested, the latter test measuring the performance of model prediction when critical detectors, which are usually located in central areas and provide crucial information for traffic control, fail. In random failure, the faulty detectors selected each time include detectors with low, medium, and high daily traffic volumes. All detectors are divided into 20 categories by their average daily traffic volume. Category 1 is the top 5% and category 20 the bottom 5% by daily traffic volume. For random failure, detectors from each category are randomly selected and set to Failure. For heavy-traffic detector failure, detectors from each layer in categories 1-5 are randomly selected and set to Failure.

The model's predictive ability would decline as an increasing percentage of detectors fail, but the performance of a good predictive model should not deteriorate drastically. In the testing, some detector failures are simulated. When a detector is set to Failure, it is assumed that its historical observations for that day are not known. The input of the model will therefore contain null values for the failed detector. The missing data of the failed detectors in each time slice are filled with the mean of the observations of working detectors in the same cluster.

Five detector failure settings are tested for each of the heavy traffic and random scenarios. For the large cities (Tokyo and Osaka) and the small city (Kochi), the failure rates were set to 1%-20% and 2%-20%, respectively. Fig. 16 compares the average value of the MAEs obtained from these tests.

The comparison shows that the MC method has a higher tolerance for failure. Without detector failure, for Tokyo, the RMSE and MAPE obtained with the MC method are only 1 veh/5 min and 1% less, respectively, than those obtained with the SC method. As the number of failures increases, the advantages of the MC method over the SC method become more pronounced. When 20% of heavy-traffic detectors fail, for Tokyo, the RMSE and MAPE obtained with the MC method are 4.7 veh/5 min and 6.1% lower, respectively, than those obtained with the SC method; for Osaka, these two indicators are 7.8 veh/5 min and 8% lower than those obtained with the SC method. The MC method is more robust against failure even on the Kochi dataset, where the SC method performs well.

In the above test, the missing values of the failed detectors for each time slice were filled with the mean of other observations in the same cluster in the same time slice (denoted by MC in Fig. 17). When predicted traffic volumes of the failed detectors are available to be used as inputs (MC(Pred) in Fig. 17), the increases in MAE and MAPE are very slight, within 0.1 veh/5 min and 0.3%, respectively, even when the failure rate is as high as 20%. This indicates that the models are more robust against failure when predicted volumes are available and substituted as inputs of failed detectors.

5.4 Setting the minimum number of detectors in a cluster

Increasing the number of clusters will result in clusters with fewer detectors. It is necessary to find a method to obtain the optimal number of clusters. In Section 5.3, we used the average traffic volume of the cluster as the model's input from the failed detectors. Hence, a cluster must have a minimum number of detectors to make it resilient to random detector failure. Setting the minimum number of detectors in each cluster to 20, 50, 100, and 200, we obtained 11 and 5 clusters (for Tokyo) for the first two and the last two minimum cluster sizes, respectively (see Table 6).

TABLE 6. Cluster size obtained for the minimum number of detectors in a cluster

Minimum cluster size	20	50	100	200
Tokyo	11	11	5	5
Osaka	18	11	7	5
Kochi	7	3	3	1

It can be seen from Fig. 18 that a large number of clusters gives slightly better prediction results. This better

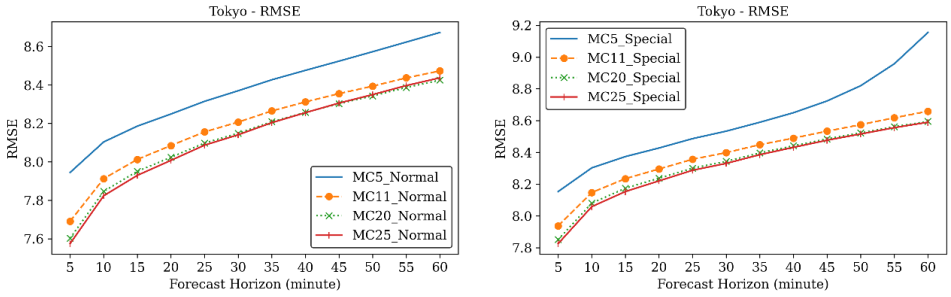


FIGURE 18. Multi-step prediction results of the multi-cluster method with different cluster sizes (without failures): Tokyo

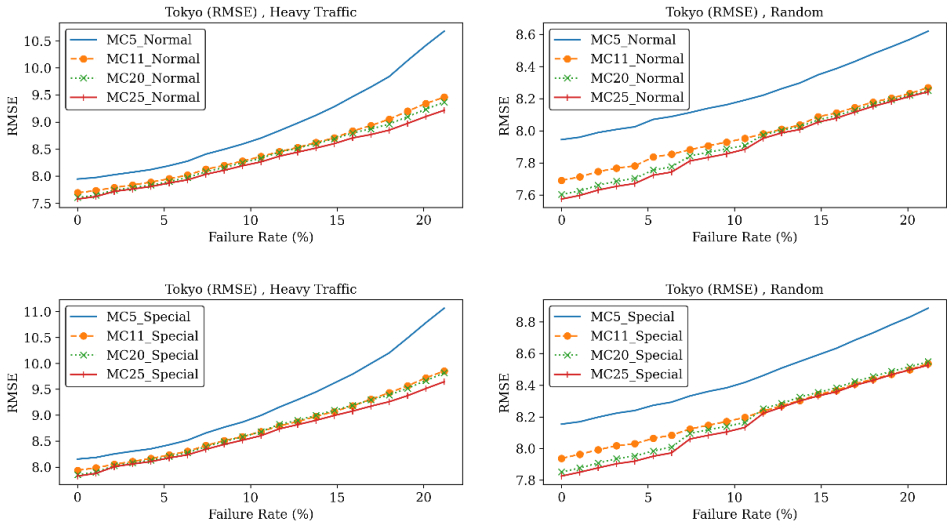


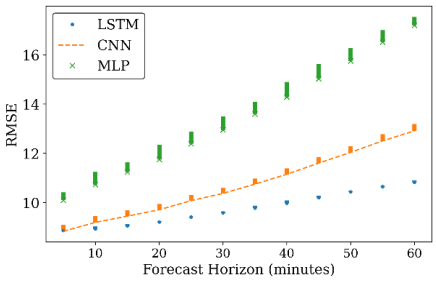
FIGURE 19. Prediction results of the multi-cluster method with different cluster sizes (with failures): Tokyo. Overall prediction for the whole network (heavy-traffic/random failures)

performance occurs because more clusters mean fewer detectors in each cluster, resulting in greater homogeneity and therefore greater ease of learning. When scenarios with detector failure are tested, the overall prediction results for the whole network are still slightly better for a large number of clusters (Fig. 19). However, clusters with a few detectors are vulnerable when detectors in these clusters fail. Hence, a trade-off between the number of clusters (homogeneity) and the minimum number of detectors in each cluster (robustness) is needed.

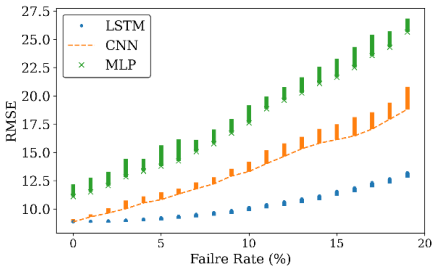
For Tokyo, the ideal minimum number of detectors in each cluster is set to 50, resulting in 11 clusters. Tests with 20 and 25 clusters show that the RMSE and MAPE of 11 clusters are only 0.1 veh/5 min and 0.2% higher, respectively. Hence, the larger number of clusters can be ignored, as the improvement in accuracy is small.

5.5 Comparison with other DL models

Fig. 20 compares the multi-step predictive performance and robustness to the failure of SC methods based on MLP and CNN. The test includes normal dates and lockdown period. The height of each bar in the figure indicates the range of RMSE obtained by the corresponding method. The LSTM is represented by the proposed SC method.



(a) Multi-step prediction



(b) Prediction with failures

FIGURE 20. Single-cluster method: Comparison of different deep learning methods. (a) Multi-step prediction (without fail- ure), (b) overall prediction for the whole network (heavy-traffic failure)

By comparing the prediction results obtained by the different settings of the number of hidden layers, CNN is set to three hidden layers, MLP one hidden layer. A ReLU activation function follows each hidden layer. In MLP, the number of units of each hidden layer is set to 2,048. In CNN, each hidden layer has 64 convolution kernels of size 3x3. The predictive ability of MLP is worse than that of CNN and LSTM because it focuses on learning global relations. The entire urban traffic network contains many heterogeneous detectors; that is, the network has many complex features. When the data of all detectors are mixed, there will be too much noise for MLP to learn the features of each type of detector. MLP can only learn features common to all detectors, such as low traffic volume at 1 am. Such features are insufficient for accurate traffic volume prediction. Moreover, MLP regards all detectors as equally important, so the performance degradation is large when there are failed detectors, that is, missing input data. Furthermore, increasing the number of hidden layers of the MLP results in serious overfitting problems. The addition of hidden layers to CNN did not bring further performance improvement either.

The predictive ability of CNN without failure is similar to that of LSTM in the 5 min time horizon. However, as the time horizon increases, its predictive ability becomes significantly inferior because LSTM is excellent in capturing temporal relationships. Moreover, MLP and CNN are not as robust as LSTM against detector failure, whether the detector is the random or heavy-traffic type. Due to the difficulty of arranging the investigated detectors to regular representations and large-scale traffic graph learning, the proposed method is currently not compared with other DL methods suitable for small areas or gridded traffic data forms.

6 Conclusion

A DL model is developed to leverage historical observations to predict network-level traffic considering detector reliability. The proposed predictive model is based on MLP and LSTM, exploiting the powerful ability of LSTM to extract temporal relationships. An MLP-based profile model is utilized to find detectors with similar signatures for each detector.

Specifically, the profile model is used to extract the most important profile from each daily data, and the signature of the detector needs to be identified from the profile. The more similar the profiles are, the more similar the travel patterns of the detectors are. Detectors with similar profiles are grouped into a cluster, and the similarity in traffic patterns causes these detectors to benefit from each other's predictions. This approach can find similar detectors without requiring additional information to build large-scale traffic graphs or compress data into regular tensors. Next, an LSTM-based predictive model is trained for each cluster.

The trade-off between the number of clusters and the cluster size is analyzed based on the model's robustness against detector failure. The methodology produces a predictive model with high accuracy. In particular, considering the MAE, RMSE, and MAPE, the adopted model for three datasets (Tokyo, Osaka, and Kochi) on normal days and special days (the COVID-19 lockdown period in Japan) and during the Tokyo Olympics were 4.5-6 veh/5 min, 6-9 veh/5 min, and 16.5%-18.5%, respectively. In conclusion, the proposed methodology is fast, accurate, and suitable for online traffic control.

In a future study, we will extend the proposed method to predict other traffic flow variables, such as speed. Further, it

will be interesting to integrate the proposed method into a proactive traffic signal control and evaluate its benefits, especially in the presence of detector failure.

ACKNOWLEDGMENT

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. R5029-18).

REFERENCES

Abduljabbar, R., & Dia, H. (2021). Short-term traffic forecasting: An lstm network for spatial-temporal speed prediction. Future Transportation, 1(1), 21-37.

Abduljabbar, R., Dia, H., Liyanage, S., & Bagloee, S. A. (2019). Applications of artificial intelligence in transport: An overview. Sustainability, 11(1).

Abduljabbar, R., Dia, H., & Tsai, P.-W. (2021). Unidirectional and bidirectional lstm models for short-term traffic prediction. Journal of Advanced Transportation, 2021, 1-16.

Albalate, D., & Fageda, X. (2019). Congestion, road safety, and the effectiveness of public policies in urban areas. Sustainability, 11(18).

Bhanu, M., Mendes-Moreira, J., & Chandra, J. (2021). Embedding traffic network characteristics using tensor for improved traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3359-3371.

Bui, K.-T. T., Torres, J. F., Gutiérrez-Avilés, D., Nhu, V.-H.,Bui, D. T., & Martínez-Álvarez, F. (2022). Deformation forecasting of a hydropower dam by hybridizing a long short-term memory deep learning network with the coronavirus optimization algorithm. Computer-Aided Civil and Infrastructure Engineering, 37(11), 1368-1386.

Chen, C., Wang, Y., Li, L., Hu, J., & Zhang, Z. (2012). The retrieval of intra-day trend and its influence on traffic prediction. Transportation Research Part C: Emerging Technologies, 22, 103-118.

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020). Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values. Transportation Research Part C: Emerging Technologies, 118, 102674.

De Boer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y.(2005). A tutorial on the cross-entropy method. Annals of Operations Research, 134, 19–67.

Deng, L., Wu, C., Lian, D., & Zhou, M. (2022). Transposed variational auto-encoder with intrinsic feature learning for traffic forecasting. arXiv preprint arXiv:2211.00641.

Diao, Z., Wang, X., Zhang, D., Liu, Y., Xie, K., & He, S. (2019). Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In Proceedings of the aaai conference on artificial intelligence (Vol. 33, p. 890-897).

Do, L. N., Vu, H. L., Vo, B. Q., Liu, Z., & Phung, D. (2019). An effective spatial-temporal attention based neural network for traffic flow prediction. Transportation research part C: emerging technologies, 108, 12–28.

Duan, Y., Yisheng, L., & Wang, F.-Y. (2016). Travel time prediction with lstm neural network. In 2016 ieee 19th international conference on intelligent transportation systems (itsc) (p. 1053-1058).

Fang, S., Zhang, Q., Meng, G., Xiang, S., & Pan, C. (2019). Gstnet: Global spatial-temporal network for traffic flow prediction. In Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI-19 (p. 2286-2293).

Fernandes, P., Coelho, M., & Rouphail, N. (2017). Assessing the impact of closely-spaced intersections on traffic operations and pollutant emissions on a corridor level. Transportation Research Part D: Transport and Environment, 54, 304-320.

Grötschla, F., & Mathys, J. (2022). Hierarchical graph structures for congestion and eta prediction. arXiv preprint arXiv:2211.11762.

Guo, K., Hu, Y., Qian, Z., Liu, H., Zhang, K., Sun, Y., . . . Yin, B. (2021). Optimized graph convolution recurrent neural network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 22(2), 1138-1149.

Guo, S., Lin, Y., Li, S., Chen, Z., & Wan, H. (2019). Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. IEEE Transactions on Intelligent Transportation Systems, 20(10), 3913–3926.

Han, L., Zheng, K., Zhao, L., Wang, X., & Shen, X. (2019). Short-term traffic prediction based on deepcluster in large-scale road networks. IEEE Transactions on Vehicular Technology, 68(12), 12301-12313.

He, P., Jiang, G., Lam, S.-K., & Tang, D. (2019). Travel-time prediction of bus journey with multiple bus trips. IEEE Transactions on Intelligent Transportation Systems, 20(11), 4192-4205.

Huang, R., Huang, C., Liu, Y., Dai, G., & Kong, W. (2020). Lsgcn: Long short-term traffic prediction with graph convolutional networks. In C. Bessiere (Ed.), Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20 (p. 2355-2361). International Joint Conferences on Artificial Intelligence Organization. Main track.

Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. IEEE Transactions on Intelligent Transportation Systems, 15(5), 2191-2201.

Huang, Z., Xia, J., Li, F., Li, Z., & Li, Q. (2019). A peak traffic congestion prediction method based on bus driving time. Entropy, 21(7).

Jia, Y., Wu, J., & Du, Y. (2016). Traffic speed prediction using deep learning method. In 2016 ieee 19th international conference on intelligent transportation systems (itsc) (p. 1217-1222).

Jiang, X., & Adeli, H. (2004). Wavelet packet-autocorrelation function method for traffic flow pattern analysis. Computer-Aided Civil and Infrastructure Engineering, 19(5), 324–337.

Johnson, S. C. (1967). Hierarchical clustering schemes. Psychometrika, 32, 241-254.

Karlaftis, M. G., & Vlahogianni, E. I. (2009). Memory properties and fractional integration in transportation time-series Transportation Research Part C: Emerging Technologies, 17(4), 444-453.

Ke, R., Li, W., Cui, Z., & Wang, Y. (2020). Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact. Transportation Research Record, 2674(4), 459-470.

Kuang, X., Xu, L., Huang, Y., & Liu, F. (2010). Real-time forecasting for short-term traffic flow based on general regression neural network. In 2010 8th world congress on intelligent control and automation (p. 2776-2780).

Lee, D., Jung, S., Cheon, Y., Kim, D., & You, S. (2018). Forecasting taxi demands with fully convolutional networks and temporal guided embedding. In Proceedings of the 32nd conference on neural information processing systems (p. 1-5).

Lefèvre, S., Sun, C., Bajcsy, R., & Laugier, C. (2014). Comparison of parametric and non-parametric approaches for vehicle speed prediction. In 2014 american control conference (p. 3494-3499).

Li, C., He, Q., & Wang, P. (2022). Estimation of railway track longitudinal irregularity using vehicle response with information compression and bayesian deep learning. Computer-Aided Civil and Infrastructure Engineering, 37(10), 1260–1276.

Li, G., Knoop, V. L., & Van Lint, H. (2021). Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations. Transportation Research Part C: Emerging Technologies, 128, 103185.

Li, J., Li, J., & Gong, Y.-J. (2022). Multi-task learning for sparse traffic forecasting. arXiv preprint arXiv:2211.09984.

Li, M., & Zhu, Z. (2021). Spatial-temporal fusion graph neural networks for traffic flow forecasting. In Proceedings of the thirty-fifth aaai conference on artificial intelligence (p. 4189-4196).

Li, T., Zhang, J., Bao, K., Liang, Y., Li, Y., & Zheng, Y. (2020). Autost: Efficient neural architecture search for spatio-temporal prediction. In Proceedings of the 26th acm sigkdd international conference on knowledge discovery amp; data mining (p. 794-802). New York, NY, USA.

Liu, Y., Lyu, C., Zhang, Y., Liu, Z., Yu, W., & Qu, X. (2021). Deeptsp: Deep traffic state prediction model based on large-scale empirical data. Communications in Transportation Research, 1, 100012.

Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., & Ji, S. (2021). Temporal multi-graph convolutional network for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3337-3348.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873.

Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., & Zhou, X. (2018). Lc-rnn: A deep learning model for traffic speed prediction. In Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18 (p. 3470-3476).

Ma, D., Song, X., & Li, P. (2021). Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns. IEEE Transactions on Intelligent Transportation Systems, 22(5), 2627-2636.

Neyshabur, B., Sedghi, H., & Zhang, C. (2020). What is being transferred in transfer learning. In Advances in neural information processing systems (Vol. 33, pp. 512–523).

Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., & Zhang, J. (2019). Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th acm sigkdd international conference on knowledge discovery amp; data mining (p. 1720–1730). New York, NY, USA: Association for Computing Machinery.

Pang, J., Huang, J., Du, Y., Yu, H., Huang, Q., & Yin, B. (2019). Learning to predict bus arrival time from heterogeneous measurements via recurrent neural network. IEEE Transactions on Intelligent Transportation Systems, 20(9), 3283-3293.

Peled, I., Lee, K., Jiang, Y., Dauwels, J., & Pereira, F. C. (2021). On the quality requirements of demand prediction for dynamic public transport. Communications in Transportation Research, 1, 100008.

Polasek, W. (2013). International Statistical Review / Revue Internationale de Statistique, 81(2), 323-325.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (p. 1135–1144). New York, NY, USA: Association for Computing Machinery.

Samant, A., & Adeli, H. (2000). Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis. Computer-Aided Civil and Infrastructure Engineering, 15(4), 241–250.

Song, Q., Ming, R., Hu, J., Niu, H., & Gao, M. (2020). Graph attention convolutional network: Spatiotemporal modeling for urban traffic prediction. In 2020 ieee 23rd international conference on intelligent transportation systems (itsc) (pp.1–6).

Song, X., Li, W., Ma, D., Wang, D., Qu, L., & Wang, Y. (2018). A match-then-predict method for daily traffic flow forecasting based on group method of data handling. Computer-Aided Civil and Infrastructure Engineering, 33(11), 982-998.

Soua, R., Koesdwiady, A., & Karray, F. (2016). Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In 2016 international joint conference on neural networks (ijcnn) (p. 3195-3202).

Tarunesh, I., & Chung, E. (2020). Predicting traffic volume and occupancy at failed detectors. Transportation Research Procedia, 48, 1072-1083. Recent Advances and Emerging Issues in Transport Research – An Editorial Note for the Selected Proceedings of WCTR 2019 Mumbai. doi:

Tedjopurnomo, D. A., Bao, Z., Zheng, B., Choudhury, F., & Qin, A. K. (2020). A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. IEEE Transactions on Knowledge and Data Engineering, 14(8), 1-20.

Treiber, M., Kesting, A., & Wilson, R. E. (2011). Reconstructing the traffic state by fusion of heterogeneous data. Computer-Aided Civil and Infrastructure Engineering, 26(6), 408-419.

Turner, S., Albert, L., Gajewski, B., & Eisele, W. (2000). Archived intelligent transportation system data quality: Preliminary analyses of san antonio transguide data. Transportation Research Record, 1719(1), 77-84.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. Transport Reviews, 24(5), 533-557.

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies, 43, 3-19. Special Issue on Short-term Traffic Flow Forecasting.

Wang, S., Huang, W., & Lo, H. K. (2019). Traffic parameters estimation for signalized intersections based on combined shockwave analysis and bayesian network. Transportation Research Part C: Emerging Technologies, 104, 22-37.

Wheelwright, S., Makridakis, S., & Hyndman, R. J. (1984, 01). Forecasting: Methods and applications. In (Vol. 35).

Williams, B. M., Durvasula, P. K., & Brown, D. E. (1998). Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models. Transportation Research Record, 1644(1), 132-141.

Wu, X., Lyu, C., Lu, Q.-L., & Mahajan, V. (2022). Similarity based feature extraction for large-scale sparse traffic forecasting. arXiv preprint arXiv:2211.07031.

Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. Transportation Research Part C: Emerging Technologies, 90, 166–180.

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. In Proceedings of the twenty-eighth international joint conference on artificial intelligence (p. 1907-1913).

Xu, Y., Lu, X., Cetiner, B., & Taciroglu, E. (2021). Real-time regional seismic damage assessment framework based on long short-term memory neural network. Computer-Aided Civil and Infrastructure Engineering, 36(4), 504-521. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12628 doi:

Yao, H., Liu, Y., Wei, Y., Tang, X., & Li, Z. (2019). Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In The world wide web conference (p. 2181-2191). New York, NY, USA.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B. (2021). Deep learning on traffic prediction: Methods, analysis and future directions. IEEE Transactions on Intelligent Transportation Systems, 1-17.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In Proceedings of the 27th international conference on neural information processing systems - volume 2 (p. 3320–3328). Cambridge, MA, USA: MIT Press.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In International joint conferences on artificial intelligence organization (p. 3634-3640).

Zang, D., Ling, J., Wei, Z., Tang, K., & Cheng, J. (2019). Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network. IEEE Transactions on Intelligent Transportation Systems, 20(10), 3700-3709.

Zhang, B., Xing, K., Cheng, X., Huang, L., & Bie, R. (2012). Traffic clustering and online traffic prediction in vehicle networks: A social influence perspective. In 2012 proceedings ieee infocom (p. 495-503).

Zhang, J., Zheng, Y., Sun, J., & Qi, D. (2020). Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Transactions on Knowledge and Data Engineering, 32(3), 468-478.

Zhang, S., Zhou, L., Chen, X., Zhang, L., Li, L., & Li, M. (2020). Network-wide traffic speed forecasting: 3d convolutional neural network with ensemble empirical mode decomposition. Computer-Aided Civil and Infrastructure Engineering, 35(10), 1132-1147.

Zhang, X., Huang, C., Xu, Y., Xia, L., Dai, P., Bo, L.,Zheng, Y. (2021). Traffic flow forecasting with spatial-temporal graph diffusion network. In Proceedings of the aaai conference on artificial intelligence (Vol. 35, p. 15008-15015).

Zheng, J., & Liu, H. (2017). Estimating traffic volumes for signalized intersections using connected vehicle data. Transportation Research Part C: Emerging Technologies, 79, 347-362.

Zheng, Z., Yang, Y., Liu, J., Dai, H.-N., & Zhang, Y. (2019). Deep and embedded learning approach for traffic flow prediction in urban

informatics. IEEE Transactions on Intelligent Transportation Systems, 20(10), 3927-3939.

Zhou, T., Huang, B., Li, R., Liu, X., & Huang, Z. (2022). An attention-based deep learning model for citywide traffic flow forecasting. International Journal of Digital Earth, 15(1), 323-344. doi: