High-performance traffic speed forecasting based on spatiotemporal clustering of road segments

Zhengchao Zhang1 | Fang He2 | Xi Lin1 | Yinhai Wang3 | Meng Li1

1 Department of Civil Engineering, Tsinghua University, Beijing, P.R. China
2 Department of Industrial Engineering, Tsinghua University, Beijing, P.R. China
3 Department of Civil and Environmental Engineering, University of Washington, Seattle, USA

Abstract
Traffic speed prediction is an indispensable element of intelligent transportation systems. Numerous studies have devoted to high-precision prediction models. However, most existing methods implement the link-wise or network-wide input. The former is time-consuming especially for large-scale applications, while the latter may incur the dilemma of underfitting owing to the heterogeneous traffic states within the entire network. Herein, we propose a novel prediction scheme based on spatiotemporal traffic pattern clustering. Firstly, road segments are partitioned into several groups via the developed clustering approach, which considers both the observed data sequence and spatial topology structure. Subsequently, sequence-to-sequence learning architecture is employed for each group to generate predictions for the entire traffic network. Validated by a real-world dataset in Beijing, our proposed paradigm offers a significant improvement over other well-known benchmarks for various prediction intervals in terms of prediction accuracy and computational efficiency.

1 | INTRODUCTION
Short-term traffic prediction is becoming increasingly essential for providing high-quality transportation services [1]. Reliable future traffic state information is conducive for formulating effective management (e.g. signal timing optimisation [2] and ramp control [3]) and assisting travellers’ route planning and departure time scheduling [4]. However, it remains a challenging task owing to the requirements of both prediction precision and timely calculation. With the rapid development of artificial intelligence technology, new prediction models have gradually emerged with deeper structures. These models offer significant improvements in accuracy performance but impose greater computation burden. Therefore, it is crucial to maintain the quality of these methods while reducing computational load significantly. Hence, we herein introduce a novel strategy that applies a deep-learning framework in classified groups of road segments, where classification is conducted to reduce the scale of learning.

The main contributions of this research can be summarised as follows:

1. A spatiotemporal clustering method of road segments is proposed, which considers both traffic speed patterns and spatial topology information.
2. Based on the clustering results, we developed a novel prediction scheme. The sequence-to-sequence learning (Seq2Seq) algorithm is separately deployed cluster by cluster in parallel.
3. We conducted numerical tests on a large-scale real-world dataset provided by AMAP, a leading mobile navigation application in China. The results indicate that our approach yields better performances than other state-of-the-art baselines in terms of both prediction accuracy and computational efficiency.

The remainder of this study is organised as follows. Section 2 presents recent relevant studies. Section 3 describes our proposed method. Section 4 presents the experimental dataset and analysis result. Finally, Section 5 concludes the study and outlines future research.

2 | LITERATURE REVIEW
In this section, we review traffic prediction models and traffic prediction with clustering technology, which are related to the current study.
2.1 Traffic prediction models

In the past decade, substantial efforts have been expended to model the temporal and spatial correlations of traffic state evolution under the topic of forecasting, the summary of which is presented as follows.

Statistical methods such as auto-regressive integrated moving average, Kalman filtering, and their variants [5–7] can predict future values based on previous observations by successive time-series analysis. However, statistical models typically rely on the stationary assumption, and they cannot accurately capture the time-varying and non-linear characteristics of traffic flow. The massive accumulation of traffic data provided opportunities for the realisation of data-driven approaches. The fuzzy theory [8], artificial neural network (ANN) [9], support vector machines [10], and K-nearest neighbour model (KNN) [11] have also been exploited for short-term traffic prediction. Furthermore, deep-learning methods have reported better performances. Recurrent neural networks (RNNs) [12], stacked auto-encoders [13], and deep belief networks [14] are powerful tools for learning temporal dependencies from data sequence.

To construct a robust prediction framework, the temporal features must be extracted and the spatial information of road networks must be incorporated. Conventional vector autoregressive models consider the effects of upstream and downstream links [15]. Cross-correlation coefficient [16] and convolutional neural networks (CNNs) [17, 18] have been utilised to analyse spatial relevance. Subsequently, graph convolution networks (GCNs) have been proposed to address non-Euclidean structured data, which have been investigated in the transportation domain. Temporal convolution networks [19], long short-term memory networks (LSTMs) [20], and Seq2Seq [21] were fused with GCNs to depict spatiotemporal dependencies in traffic prediction.

To summarise, remarkable achievements of various models have been reported in previous studies. Unlike the aforementioned studies, we propose an efficient and general prediction paradigm based on the clustering of road segments.

2.2 Traffic prediction with clustering technology

In [22], a non-parametric clustering-based technique that provides accurate traffic forecasting was established through the exploitation of traffic data dynamics, whereas in [23], an urban traffic flow prediction system using a multifactor pattern recognition model was established, which combined a Gaussian mixture model clustered with an ANN. Tang et al. constructed a fuzzy neural network to forecast travel speed, which was trained by the K-means method to partition inputs into different clusters and a weighted recursive least-squares estimator to optimise the parameters [24]. Subsequently, Wang et al. developed a traffic threshold identification module based on an improved K-means clustering algorithm and implemented a traffic flow prediction framework based on adaptive speed thresholds [25]. Shen et al. presented a novel traffic speed prediction method based on temporal clustering analysis and a deformable convolution neural network that could discriminate traffic environments [26]. The aforementioned clustering targets were the observed data samples, where the spatial information of road networks was not incorporated.

Song et al. measured the similarities between road segments based on adjacency, connectivity, and congestion; subsequently, they generated topological maps by road segment clustering [27]. However, the way to proceed with traffic prediction was not involved. The spatiotemporal performance trends at the network level and for individual links were mined through K-means clustering [28], which could create new prediction error measures rather than enhance the prediction performance. In [29], a clustering procedure was used to identify time-variant clusters of road segments with high spatial correlations; subsequently, they were fed into an ANN-based traffic forecasting model, which is the most correlated model with ours. Nevertheless, the simple multilayer perceptron (MLP) structure will impair the capacity in capturing the temporal dependencies, especially for multistep prediction. We further propose obtaining the optimal number of clusters through the pre-experiment on a validation set along with providing a comprehensive comparison of prediction performance between link-, cluster-, and network-level inputs.

3 METHODOLOGY

In this section, we first interpret the notations of the variables used herein; subsequently, we illustrate the devised method in detail.

3.1 Preliminaries

We model the network as a directed graph $G(\mathcal{N}, \mathcal{L})$, where the node-set $\mathcal{N}$ comprises road segments (also called links), and $\mathcal{L}$ is the set of directed edges representing the connections among the road segments. $D$ the distance matrix of the road segments and the variable $D_{ij}$ denotes the length of the shortest path from the road segment $x_i$ to $x_j$ considering the driving direction.

We denote $\dot{v}_t$ as the speed of the road segment $x_t$ ($\forall x_t \in \mathcal{N}$) at the $t^{th}$ time slot, where the time slot is divided with a fixed duration (e.g., 5 min), and the ‘speed’ is defined as the average speed of floating cars travelling through the road segment during this time interval. $V_t \in \mathbb{R}^{\mathcal{N}} (|\mathcal{N}| = N$ the total number of road segments) is the speed vector of the underlying road network at the $t^{th}$ time slot, where the $t^{th}$ element is $(V_t)_i = \dot{v}_t$.

Given the historical observations $\{V_{t-s} \mid s = 0, 1, \ldots, m\}$ ($m$ is the look-back time window) and network topology $G$, the traffic speed forecasting problem is to seek the trustworthy approximation $\hat{V}_{t+s}$ of the future value $V_{t+s}$, where $n$ is the required prediction step.
3.2 Road segment clustering

The traffic conditions of the entire road network vary significantly over both time and space [30], which is challenging for traffic prediction. However, focusing on the local part, the traffic states are relatively stable and closely dependent [31]. Hence, our major objective is to partition the given road network into clusters, with each containing a group of considerably correlated road segments.

Based on the above notations, we define \( P_{ij} \) to characterise the similarity in traffic patterns between road segments \( x_i \) and \( x_j \):

\[
P_{ij} = \frac{\sum_n (\alpha - \gamma_n)^2}{|\tau|}, \quad n \in \tau,
\]

where \( \tau \) refers to a historical observed dataset that covers a sufficient period (e.g., one week), and \(|\tau|\) denotes its cardinality.

However, only considering the observed data into account may yield ambiguous results for the sake of coincidences in numerical values (e.g., two irrelevant road segments may possess similar speeds). As the traffic flow at downstream and upstream roads affect each other through transfer and feedback effects [32], it is necessary to incorporate the topology information into the clustering procedure. Combined with the spatial distance, the spatiotemporal similarity between road segments \( i \) and \( j \) can be quantified as

\[
S_{ij} = \alpha P_{ij} + \beta D_{ij},
\]

where \( P_{ij} \in [0,1] \) and \( D_{ij} \in [0,1] \) are the normalised results of \( P_{ij} \) and \( D_{ij} \), respectively; \( \alpha \) and \( \beta \) are coefficients to adjust the effects of corresponding components.

Once the similarity matrix of road segments is derived, our next goal is to partition the road segments into a set of disjoint clusters in which the total dependencies between road segments within the same cluster are maximised. Let \( P_k \) be the \( k \textsuperscript{th} \) cluster with \( q \) road segments and \( M \) be the total number of clusters:

\[
P_k = \arg \min_{x_1, x_2, ..., x_q} \left\{ \frac{\sum_j \sum_i S_{ij}}{q} \mid x_i, x_j \in \mathcal{N} \right\}
\]

\[
\text{s.t.} \left\{ \begin{array}{l}
P_k \cap P_w = \emptyset, \forall k, w \in [1, ..., M] \\
\bigcup_{k=1}^M P_k = \mathcal{N}
\end{array} \right. \]  \tag{3}

This can be solved via agglomerative clustering, which generates clusters in a bottom-up way to produce a binary tree with each data point as either a leaf node or root node [33].

The algorithm is initialised by treating each road segment as a separate cluster. The second step is to calculate a distance matrix embodying the distance between any two clusters. In this study, we used the average-linkage criterion [34], which determines the average similarity between all road segments of the two clusters as the distance for clustering. It is expressed as

\[
R_{ij} = \frac{\sum_{x_i \in P_j} s_{ij}}{|P_i||P_j|}, \quad x_i \in P_j, x_j \in P_j
\]  \tag{4}

In the third step, the two clusters with the shortest distance are identified and merged into a new cluster. Subsequently, steps 2 and 3 are repeated until the number of clusters reaches the given number.

Owing to the clustering processing, the road segments in the underlying network are classified into several groups. In the following sections, we denote \( \{x_{k_1}, x_{k_2}, ..., x_{k_q}\} \) as the road segments in the cluster \( P_k \), and \( V_i^{(k)} = \{v_i^{k_1}, v_i^{k_2}, ..., v_i^{k_q}\} \) as the corresponding speed observations at the \( i \textsuperscript{th} \) time slot.

3.3 Traffic prediction method

As road segments within the same cluster are strongly spatial correlated, the primary objective of the traffic prediction model is to capture the temporal evolution of the traffic state. We leverage the Seq2Seq architecture [35] to model the temporal dependencies, which unites two RNN modules to overcome the disadvantage of fixed output timestamp of a single RNN. As shown in Figure 1, the encoder embeds the input sequence \( \{V_{t-1}^{(k)}, ..., V_{t}^{(k)}\} \), and its final hidden state is fed into the decoder, which learns to predict the future traffic speed \( \hat{V}_{t+1}^{(k)}, \hat{V}_{t+2}^{(k)}, ..., \hat{V}_{t+h}^{(k)} \). It is noteworthy that the data stream of our Seq2Seq model is cluster-level \( V_i^{(k)} \), which is vastly different from link-level \( v_i^{k} \) and network-level \( V_i \) in previous studies [36]. The benefits of this manner will be emphasised in Section 4.

During training, historical ground truth series are fed to the decoder (see Figure 1(a)), whereas ground truth observations are replaced by predictions generated by the model itself during testing (see Figure 1(b)). The model performance may degrade due to the discrepancy of input distributions between the training and testing stage. Hence, scheduled sampling is integrated [37] into the model, where the encoder is fed with either the ground truth observation with probability \( \xi_i \) or the prediction by the model with probability 1-\( \xi_i \) at the \( i \textsuperscript{th} \) iteration. In this study, we set \( \xi_i \) as

\[
\xi_i = \begin{cases} 1, & i < \Lambda \\ 0, & i \geq \Lambda \end{cases}
\]  \tag{5}

where \( \Lambda \) is the predefined threshold.

Throughout the model operation, the hidden state \( h_{t-1} \), which is passed to next timestamp \( t \), together with input \( V_{t-1}^{(k)} \) generates a new hidden state \( h_t \) at each iteration. Both the encoder and decoder adopt the LSTM structure [38], and the calculation formulas are shown in Equations (6)–(11):

\[
f_t = \sigma(W_f \cdot [h_{t-1} || V_i^{(k)}] + b_f)
\]  \tag{6}

\[
i_t = \sigma(W_i \cdot [h_{t-1} || V_i^{(k)}] + b_i)
\]  \tag{7}
where $f_t$, $i_t$, $o_t$, and $c_t$ represent the forget gate, input gate, output gate, and memory cell vector, respectively; $\mathbf{[\cdot|\cdot]}$ concatenates two tensors along the same dimensions; $\sigma(y) = \frac{1}{1+e^y}$ and $\tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$ are two widely used non-linear activation functions that map the input into (0,1) and $(-1,1)$, respectively; $\odot$ refers to the Hadamard product that conducts the element-wise multiplication operation; $W_f$, $W_i$, $W_c$, and $W_o$ are weight matrices that conduct a linear transformation while $b_f$, $b_i$, $b_c$, and $b_o$ are the corresponding bias vectors.

Equation (12) denotes the fully connected layer from the hidden state to the output. The dimensions of the weighted parameter matrix $W_v$ and intercept parameter $b_v$ are consistent with the output:

$$\hat{V}^{(k)}_{t+j} = W_v h_{t+j-1} + b_v, \quad 1 \leq j \leq n$$  (12)

To jointly reduce the errors in the multiple-step prediction, the loss function is defined as the mean absolute error between $(\hat{V}^{(k)}_{t+1}, \hat{V}^{(k)}_{t+2}, ..., \hat{V}^{(k)}_{t+n})$ and $(V^{(k)}_{t+1}, V^{(k)}_{t+2}, ..., V^{(k)}_{t+n})$, which is given by

$$\text{loss} = \frac{1}{n} \sum_{j=1}^{n} \|\hat{V}^{(k)}_{t+j} - V^{(k)}_{t+j}\|_1$$  (13)

where $\| \cdot \|_1$ returns the $L_1$ norm of the vector.

All the parameters are updated by minimising the loss function through the stochastic mini-batch gradient descent (BGD) algorithm in the training stage. The training steps of the proposed model is elaborated in Algorithm 1.

4 | EXPERIMENTS

4.1 | Dataset description

Real-world traffic dataset was collected from anonymous navigation users of the AMAP (https://www.amap.com/). We selected the inner fourth ring road in Beijing as the study site, which encircled the downtown area. As shown in Figure 2, the 65-km fourth ring road is partitioned into 162 road segments of length 400 m. Subsequently, we calculated the 5-min average speed for each road segment by utilising the instantaneous state of the trajectory points. The missing values were completed by the nearest non-missing values using MATLAB [39]. The average traffic speeds of morning and evening peak hours on weekdays of October 2016 are shown in Figures 3(a) and (b), respectively. The entire dataset ranging from October 1, 2016, to November 27, 2016, was separated into three independent subsets, the details of which are provided in Table 1.

4.2 | Experimental setting

In this subsection, we compare our prediction scheme with several prevailing methods:
**FIGURE 2** Sketch of road segments in geographic information system map

**ALGORITHM 1: Training the model**

**Input** Traffic speed observations of $k^{th}$ cluster in the training set: $\{ V^{(k)}_1, \ldots, V^{(k)}_T \}$

Input timestamp: $m$; Prediction timestamp: $n$

Threshold of train steps for scheduled sampling: $\Lambda$

**Output** The model with learnt parameters

1. **Procedure** train Seq2Seq model
   2. Initialise two null sets: $U' \leftarrow \emptyset, V \leftarrow \emptyset$
   3. for all available time slots $i (1 \leq i \leq T)$ do
   4. $X_{\text{encoder}} \leftarrow \{ V^{(k)}_i, \ldots, V^{(k)}_T \}$, $X_{\text{decoder}} \leftarrow \{ V^{(k)}_{i+1}, \ldots, V^{(k)}_{i+n} \}$
   5. $Y_{\text{label}} \leftarrow \{ V^{(k)}_{i+1}, \ldots, V^{(k)}_{i+n} \}$
   6. $U' \leftarrow (X_{\text{encoder}}, X_{\text{decoder}}, Y_{\text{label}})$, $V \leftarrow (X_{\text{encoder}}, Y_{\text{label}})$
   7. for all training steps $i (1 \leq i \leq I)$ do
   8. if $i < \Lambda$
   9. Randomly extract a batch of samples $U^b$ from $U'$
   10. Minimise Equation (13) to update the parameters by the BGD algorithm within $U^b$
   11. else
   12. Randomly extract a batch of samples $V^b$ from $V$
   13. Minimise Equation (13) to update the parameters by the BGD algorithm within $V^b$
   14. end for
   15. end for

2. **Historical average (HA):** The HA model performs predictions based on the mean values of the same timestamp in the training set, which serves as a basic criterion.

3. **MLP:** An MLP [40] with one hidden layer is established, and we set the number of hidden neurons to be twice the feature dimension.

4. **KNN:** The KNN [41] calculates the weighted summation of the corresponding future values belonging to similar samples. The hyperparameter $K$ is selected from 5 to 30 through the performances on the validation set.

5. **LSTM:** In LSTM [38], the inputs are reshaped into a matrix with one axis as the time step and another axis as the number of road segments, which can address temporal correlations well.

6. **Graph attention network (GAT):** The GAT [42] leverages masked self-attentional layers to specify different weights to different neighbourhoods, which considers spatial information.

7. **Seq2Seq:** It is facilitated with the same structure as that of our prediction model except for the cluster-level input.

8. Traffic graph convolutional (TGC)-LSTM: The TGC-LSTM [20] unifies the traffic graph convolution operation and LSTM structure to attain spatiotemporal correlations in the transportation system.

It is noteworthy that KNN, LSTM, GAT, and Seq2Seq are applicable for both link-wise modelling and network-wide inputs. The performances of these models under the two scenarios were tested in the following experiments.

Speed records in the past 60 min (i.e. $m = 11$) were employed as inputs for all models (excluding HA) to forecast the traffic states in the next 10, 20, and 30 min. The prediction accuracy of all road segments was assessed via three typical metrics, that is, mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE), expressed as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{N} \sum_{t \in H} |\hat{v}_{it} - v_{it}|}{N \times |H|}$$

(14)

$$\text{MAPE} = \frac{\sum_{i=1}^{N} \sum_{t \in H} |\hat{v}_{it} - v_{it}|}{N \times |H|}$$

(15)

$$\text{RMSE} = \frac{\sum_{i=1}^{N} \sqrt{\sum_{t \in H} (\hat{v}_{it} - v_{it})^2}}{N}$$

(16)

where $v_{it}$ and $\hat{v}_{it}$ are the ground truth and prediction values at the $i^{th}$ road segment at the $t^{th}$ time slot, respectively; $H$ denotes the testing set.

To ensure fairness in both accuracy and efficiency comparisons, the same training hyperparameters were retained in the neural-network-based methods (except TGC-LSTM). In detail, the batch size was 512, the learning rate $10^{-2}$, and the training steps 2500 with the Adam optimiser. We accommodate the parameters of TGC-LSTM to the underlying dataset on the basis of suggestions in [20]. The LSTM, encoder, and decoder of Seq2Seq comprised of one hidden layer and eight hidden units for the link-level input and 16 hidden units for the
TABLE 1  Information of experimental datasets

| Item               | Training set | Validation set | Testing set  |
|--------------------|--------------|----------------|--------------|
| Date               | Oct 1–Nov 13 | Nov 14–Nov 20  | Nov 21–Nov 27|
| Prediction time range | 06:00–22:00 (192 data points per day) |                |              |
| Dataset size       | $44 \times 192 \times 162 \approx 1.4 \text{ M}$ | $7 \times 192 \times 162 \approx 220 \text{ K}$ | $7 \times 192 \times 162 \approx 220 \text{ K}$ |

FIGURE 3  Average traffic speed of the fourth ring road on weekdays in October 2016 (a) morning peak hours, (b) evening peak hours

network-level input. The threshold of scheduled sampling for the Seq2Seq model was $\Lambda = 1700$; 1-hop and 2-hop neighbours were considered separately in the GAT. The above settings are determined through pre-experiment on validation set which achieves a balance between prediction accuracy and computation efficiency.

For our prediction paradigm, we set $\alpha = \beta = 0.5$ and used the historical observations from October 10 to 16 for clustering (i.e. we calculated $P_{ij}$ using Equation 1). We exploit the data from one week for clustering because traffic patterns in both weekdays and weekends are appropriately considered. The optimal number of clusters was selected as 12 via the pre-experiment on the validation set of the corresponding prediction performances from (4, 8, 12, 16, 20), (the details will be provided in Section 4.4). The model settings remained the same as those mentioned above except that 16 hidden units were used.

Our experimental platform was a server with two central processing units (CPUs) (Intel(R) Xeon(R) CPU E5-2673 v3 @2.40 GHz, 24 cores), 256-GB RAM (Random Access Memory), and four Graphics Processing Units (GPUs) (NVIDIA Quadro P5000, 16 GB memory). We used python 3.6.3 with scikit-learn [43] and tensorflow [44] to achieve contrasting models. All the algorithms were coded in parallel.

4.3  Prediction performance

Tables 2–4 present the comparisons of the proposed model and benchmark algorithms for 10-, 20-, and 30-min ahead forecasting on the testing datasets. The following conclusions were obtained from the results:

(i) Our prediction strategy, that is, implementing the Seq2Seq model cluster by cluster achieved the best prediction results for all three evaluation metrics under various steps and exhibited superior computational efficiency. Although the specialised spatiotemporal deep-learning framework, TGC-LSTM, presents better performances than other baselines, it is also inferior to our approach.

(ii) Figure 4 compares the execution time and MAE of the KNN, LSTM, GAT, and Seq2Seq fed with link- and network-level inputs under a 20-min time period. As expected, link-wise modelling requires a lengthy calculation time while affording better precision, whereas the network-level input performs worse as the heterogeneous traffic states of the entire network cause the underfitting problem.

(iii) Figure 5 shows the execution time and MAE of the Seq2Seq model with link-, network-, and cluster-level inputs during a 10–30 min interval. It is clear that the clustering pre-process improved the accuracy and efficiency significantly. Compared with the link-level input, the cluster-level input incorporates spatial features and reduces the required number of training rounds from the number of links to the number of clusters (e.g. for the road network with 160 links, the link-level input needs 160 training rounds link by link. When we divide them into 20 clusters, only 20 training rounds are required). Unlike the network-level input, the cluster algorithm partitions the network...
into highly correlated groups, which enables the spatiotemporal dependencies to be learned more easily. It is noteworthy that the running time of the Seq2Seq model increases rapidly with the input dimension. The cluster-level input decreases the input dimension at each iteration, and the parallel computing for each cluster contributes to reduced time consumption. The results serve as a powerful support to our motivation of this study.

4.4 Clustering result

Figure 6 shows the change of MAE with the number of clusters in (4, 8, 12, 16, 20) on the validation and testing set under 20-min prediction step, respectively. The variation in errors exhibits the same trend between validation and testing, indicating that it is appropriate to select the optimal number of clusters through pre-experiment on the validation set. Meanwhile, the
TABLE 5  Contrast of prediction performance for clustering and random selection

| Interval | Model             | MAPE   | MAE   | RMSE  |
|----------|-------------------|--------|-------|-------|
| 10-min   | Seq2Seq(cluster)  | 10.62% | 3.99  | 6.33  |
|          | Seq2Seq(random)   | 11.82% | 4.32  | 7.04  |
| 20-min   | Seq2Seq(cluster)  | 15.01% | 5.23  | 8.68  |
|          | Seq2Seq(random)   | 16.55% | 5.65  | 9.35  |
| 30-min   | Seq2Seq(cluster)  | 18.97% | 6.21  | 10.36 |
|          | Seq2Seq(random)   | 20.40% | 6.59  | 10.75 |

results are insensitive to the number of clusters, which verifies
the robustness of our prediction scheme. Figure 7 presents the
partition results of road segments with 12 clusters, for example,
the road segments 6-13 shown in Figure 2 are divided into
cluster 2 (shown in orange) in Figure 7(a).

Comparing Figures 7(a) and (b), the distribution of clusters
without spatial information is disorderly (e.g. the road segments
in the north and south fourth ring road are both classified into
cluster 7, which unlikely owns the physical interactions), which
is clearly not the desirable partition for traffic prediction.

4.5  Discussion of cluster-level input

To further validate the effect of our prediction strategy, an
additional experiment was conducted to extract road segments
randomly to form the same number of groups as that of our
clustering process. As shown in Table 5, feeding the model with
random sets of road segments does not improve the predic-
tion accuracy, which proves the necessity of our designed clus-
tering algorithm. The results of this experiment show that the
improvements in the cluster-level input are owing to the highly
relevant spatial information within each cluster. A simple modi-
fication of the input dimension through feeding the random link
sets cannot boost the model performance.

4.6  Analysis of spatiotemporal similarity
measurement

Figure 8 depicts the performances of the proposed prediction
scheme with different coefficients $\alpha, \beta$ in Equation (2).
The results explicate that the clustering with a relative trade-
off between pattern similarity ($\tilde{P}_{ij}$) and spatial distance ($\tilde{D}_{ij}$)
is better for the prediction accuracy. The partition of road
segments without the consideration of pattern similarity ($\alpha = 0$)
or spatial distance ($\beta = 0$) will compromise the prediction
performance. Moreover, Table 6 assesses the effects of clus-
tering with our spatiotemporal similarity measurement $s_{ij}$ and
Pearson product-moment correlation coefficient [45, 46] on
prediction performances. The results prove that the devised
spatiotemporal similarity measurement is more beneficial for
promoting traffic prediction accuracy.
CONCLUSION

A novel high-performance traffic prediction paradigm was developed in this study. In the first step, road segments in the underlying road network were partitioned into several groups through a spatiotemporal clustering algorithm. Subsequently, strongly correlated and mutually influencing road segments were introduced into the same cluster. Next, the Seq2Seq model was employed cluster by cluster to perform traffic prediction. Using a real-world traffic dataset, our approach demonstrated significant improvements over other benchmarks in terms of both prediction accuracy and computational efficiency. Furthermore, we analysed the differences between link- and network-level inputs, the influences of the cluster number, the effects of randomly selected road segment groups, and the designed spatiotemporal similarity measurement. In future studies, we may extend our prediction scheme for application to large-scale urban road networks, where the influences of traffic lights may be considered into the clustering process.

ACKNOWLEDGMENTS

The research is supported by grants from the National Key R&D Program of China (2018YFB1601600). The research is supported in part by Tsinghua-Toyota Joint Research Institution.

REFERENCES

1. Lin, Y., Wang, P., Ma, M.: Intelligent Transportation System (Its): Concept, Challenge and Opportunity. In: 2017 IEEE 3rd International Conference on Big Data Security on Cloud (Bigdatasecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), Beijing, China (2017)
2. Ahmed, A., et al.: Real-time dynamic traffic control based on traffic-state estimation. Transp. Res. C 267(3), 584–595 (2019)

3. Goatin, P., Göttlich, S., Kolt, O.: Speed limit and radar meter control for traffic flow networks. Eng. Optim. 48(7), 1121–1144 (2016)

4. Yuan, J., et al.: T-drive: Enhancing driving directions with taxi drivers’ intelligence. IEEE Trans. Knowl. Data Eng. 25(1), 220–232 (2013)

5. Wang, H., et al.: A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework. Transportmetrica B Transp. Dyn. 4(3), 159–186 (2016)

6. Wang, Y., Papageorgiou, M.: Real-time freeway traffic state estimation based on extended kalman filter: a general approach. Transp. Res. Part B Methodol. 39(2), 141–167 (2005)

7. Wang, Y., Papageorgiou, M., Messmer, A.: Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing. Transp. Res. Part A Policy Pract. 42(10), 1340–1358 (2008)

8. Guo, J., et al.: Short-term traffic flow prediction using fuzzy information granulation approach under different time intervals. IET Intel. Transport Syst. 12(2), 143–150 (2017)

9. Chan, K.Y., et al.: Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. IEEE Trans. Intell. Transp. Syst. 13(2), 644–654 (2011)

10. Wang, J., Shi, Q.: Short-term traffic flow forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory. Transp. Res. Part C Emerging Technol. 27, 219–232 (2013)

11. Sun, B., et al.: Short-term traffic forecasting using self-adjusting K-nearest neighbours. IET Intel. Transport Syst. 12(1), 41–48 (2017)

12. Zhao, Z., et al.: LSTM network: A deep learning approach for short-term traffic forecast. IET Intel. Transport Syst. 11(2), 68–75 (2017)

13. Lv, Y., et al.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16(2), 865–873 (2014)

14. Huang, W., et al.: Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15(5), 2191–2201 (2014)

15. Chandra, S.R., Al-Deek, H.: Predictions of freeway traffic speeds and volumes using vector autoregressive models. J. Intell. Transp. Syst. 13(2), 53–72 (2009)

16. Pan, T., et al.: Short-term traffic state prediction based on temporal–spatial correlation. IEEE Trans. Intell. Transp. Syst. 14(3), 1242–1254 (2013)

17. Ma, X., et al.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17(4), 818 (2017)

18. Ke, J., et al.: Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. Transp. Res. Part C Emerging Technol. 85, 591–608 (2017)

19. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm (2018) <https://www.ijcai.org/Proceedings/2018/0099>

20. Cai, Z., et al.: Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Trans. Intell. Transp. Syst. 21(11), 4883–4894 (2020)

21. Zhang, Z., et al.: Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. Transp. Res. Part C Emerging Technol. 105, 297–322 (2019)

22. Kehagias, D., Salamantis, A., Tzovaras, D.: Speed pattern recognition technique for short-term traffic forecasting based on traffic dynamics. IET Intel. Transport Syst. 9(6), 646–653 (2015)

23. Kim, Y.-J., Hong, J.-S.: Urban traffic flow prediction system using a multifactor pattern recognition model. IEEE Trans. Intell. Transp. Syst. 16(5), 2744–2755 (2015)

24. Tang, J., et al.: An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. IEEE Trans. Intell. Transp. Syst. 18(9), 2340–2350 (2017)

25. Xiangxue, W., Liumai, X.: Wavelet-based short-term forecasting with improved threshold recognition for urban expressway traffic conditions. IET Intel. Transport Syst. 12(6), 463–473 (2018)

26. Shen, G., et al.: Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (May, 2018). IEEE Access 6, 51756–51765 (2018)

27. Song, Q., et al.: An urban topological map generation method for traffic flow prediction based on road segment clustering with floating vehicle trajectory dataset. In: 19th COTA International Conference of Transportation Professionals (CICTP 2019), Nanjing , China (2019)

28. Asif, M.T., et al.: Spatiotemporal patterns in large-scale traffic speed prediction. IEEE Trans. Intell. Transp. Syst. 15(2), 794–804 (2013)

29. Zhang, B., et al.: Traffic clustering and online traffic prediction in vehicle networks: a social influence perspective. In: 2012 Proceedings IEEE INFOCOM, Orlando, Florida (2012)

30. Statthopoulos, A., Karlaftis, M.: Temporal and spatial variations of real-time traffic data in urban areas. Transp. Res. Rec. 1768(1), 135–140 (2001)

31. Zou, H., et al.: A spatial analysis approach for describing spatial pattern of urban traffic state. In: 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal (2010)

32. Dong, C.-J., et al.: Spatial and temporal characteristics for congested traffic on urban expressway. J. Beihug. Univ. Technol. 38(8), 1242–1246 (2012)

33. Xia, J., Huang, W., Guo, J.: A clustering approach to online freeway traffic state identification using its data. KSCE J. Civ. Eng. 16(3), 426–432 (2012)

34. Lukasová, A.: Hierarchical agglomerative clustering procedure. Pattern Recognit. 11(5-6), 365–381 (1979)

35. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, Montreal, Canada (2014)

36. Liao, B., et al.: Deep sequence learning with auxiliary information for traffic prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK (2018)

37. Bengio, Y., et al.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems, Montreal, Canada (2015)

38. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997)

39. Fochl-Fortunya, A., Artega, F., Ferrer, A.: Missing data imputation toolbox for matlab. Chemom. Intell. Lab. Syst. 154, 93–100 (2016)

40. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature 323(6088), 533–536 (1986)

41. Dudani, S.A.: The distance-weighted K-nearest-neighbor rule. IEEE Trans. Syst. Man Cybern. SMC-6(4), 325–327 (1976)

42. Zhang, K., et al.: Graph attention temporal convolutional network for traffic speed forecasting on road networks. Transportmetrica B Transp. Dyn. 1–19 (2020).

43. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830 (2011)

44. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: Advances in Neural Information Processing Systems, Montreal, Canada (2015)

45. Cheng, T., Hasworth, J., Wang, J.: Spatio-temporal autocorrelation of road network data. J. Geog. Syst. 14(4), 389–413 (2012)

46. Diamantopoulos, T., et al.: Investigating the effect of global metrics in traffic forecast. IET Intel. Transport Syst. 11(2), 68–75 (2017)