

Research Article

Highway Travel Time Prediction of Segments Based on ANPR Data considering Traffic Diversion

Wenjun Du,¹ Bo Sun,² Jiating Kuai,¹ Jiemin Xie,³ Jie Yu,¹ and Tuo Sun^{ID}⁴

¹Zhejiang Institute of Communications Co., Ltd, Zhejiang, China

²Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

³Department of Civil Engineering, The University of Hong Kong, Hong Kong, China

⁴Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai, China

Correspondence should be addressed to Tuo Sun; suntuo@tongji.edu.cn

Received 29 May 2021; Revised 28 June 2021; Accepted 1 July 2021; Published 12 July 2021

Academic Editor: Qi-zhou Hu

Copyright © 2021 Wenjun Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Travel time is one of the most critical parameters in proactive traffic management and the deployment of advanced traveler information systems. This paper proposes a hybrid model named LSTM-CNN for predicting the travel time of highways by integrating the long short-term memory (LSTM) and the convolutional neural networks (CNNs) with the attention mechanism and the residual network. The highway is divided into multiple segments by considering the traffic diversion and the relative location of automatic number plate recognition (ANPR). There are four steps in this hybrid approach. First, the average travel time of each segment in each interval is calculated from ANPR and fed into LSTM in the form of a multidimensional array. Second, the attention mechanism is adopted to combine the hidden layer of LSTM with dynamic temporal weights. Third, the residual network is introduced to increase the network depth and overcome the vanishing gradient problem, which consists of three pairs of one-dimensional convolutional layers (Conv1D) and batch normalization (BatchNorm) with the rectified linear unit (ReLU) as the activation function. Finally, a series of Conv1D layers is connected to extract features further and reduce dimensionality. The proposed LSTM-CNN approach is tested on the three-month ANPR data of a real-world 39.25 km highway with four pairs of ANPR detectors of the uplink and downlink, Zhejiang, China. The experimental results indicate that LSTM-CNN learns spatial, temporal, and depth information better than the state-of-the-art traffic forecasting models, so LSTM-CNN can predict more accurate travel time. Moreover, LSTM-CNN outperforms the state-of-the-art methods in nonrecurrent prediction, multistep-ahead prediction, and long-term prediction. LSTM-CNN is a promising model with scalability and portability for highway traffic prediction and can be further extended to improve the performance of the advanced traffic management system (ATMS) and advanced traffic information system (ATIS).

1. Introduction

With the increase of the cooperative vehicle infrastructure system (CVIS), the informationization of highways tends to be significant for the construction of smart highways. Many pilot projects of smart highways have been proposed in China, such as 116 km of Yanchong Highway (Beijing), 248 km of Huhangyong Highway, 161 km of Hangshaoyong Highway (Zhejiang), and 885 km of Hunchun-Wulanhao Highway. ITS, which generally aims at promoting effective urban planning, route decision, and other traffic applications, requires accurate traffic travel time prediction.

However, it is a well-known challenging task, especially when involving the trip travel time distribution for an arbitrary origin-destination (OD) pair. Existing travel time estimation approaches do not fully explore the data pattern of each segment that can help improve the traffic congestion problem. For example, many deep-learning-based models have been recently proposed, in which the traffic data are processed in the form of time series. Some simple convolutional neural networks (CNNs) of feature extraction only generate partial level features and ignore the relationship of vehicles entering or exiting segments that may affect prediction performance. In order to learn segment

temporal patterns, other methods directly stack a series of images in chronological order [1, 2]. For instance, simple long short-term memory (LSTM) learns the regular temporal representation in nonlinear traffic flow data [3] and captures the inherent characteristics of long-term dependencies in sequential data. These characteristics make it a suitable choice in traffic prediction. Liu et al. [4] combined convolution and LSTM to form a Conv-LSTM module, which extracts relatively complex road information for traffic flow prediction.

Both CNNs and LSTM have succeeded in the field of image processing and sequence prediction. In the trend of traffic prediction methods, more and more people have adopted CNNs and LSTM modules. But they often neglect the characteristics of different segments and the effective use of models. In order to solve the aforementioned problems, we propose a hybrid model entitled LSTM-CNN. In our model, we use the LSTM module to capture travel correlations to enhance the relationship and expand feature propagation at the CNN layer. We apply residual network [5, 6] to the CNN layer to make it easier to train deep networks. Attention mechanism [7, 8] is applied to assign different levels of attention features based on prediction target to discover potential dependencies of highway data. By combining these technologies and mechanisms, this study will explore the potential of travel time features of automatic number plate recognition (ANPR) data with consideration of traffic diversion and other relevant information, which contributes to the management of the smart city.

The main contributions of this paper are summarized as follows. First, a large number of studies about the highway problems are only based on the location of detectors to obtain traffic data. But on the highway, due to the high cost, the distance between adjacent detectors is often far, so it is impossible to obtain a more detailed traffic state directly. It is necessary to introduce a generalized extended-segment data acquirement mode for the highway to predict accurately and find out the source of congestion, and we divide the segment into different sections based on ANPR. Second, the residual network layers (ResNet) are proposed to dig out deep features of spatiotemporal highway segments, which consist of three pairs of 1D convolutional layers and the BatchNorm, followed by a rectified linear unit (ReLU) activation layer. Third, to smooth the noisy data, we applied the method of winsorization. Fourth, a four-step mechanism combining with a hybrid spatiotemporal model can achieve the multidimensional final prediction.

The remainder of the paper is organized as follows. Section 2 summarizes the present development of the travel time prediction, while in Section 3, the methodology framework of travel time prediction is proposed, and its details are described. In Section 4, training and test data are presented, as well as the analysis of the data. In Section 5, different methods are compared with the proposed LSTM-CNN in terms of the accuracies based on the case study in this study. Finally, conclusions are given in the last section.

2. Literature Review

Travel time information is a basic component of advanced traveler information systems, which is currently the main research of ITS. This research is driven by the development of information collection and communication technologies that improve the accessibility of various types of traffic data. For example, ANPR data can be used to model the key factors that contribute to the travel time variations, such as the entry and exit of vehicles and traffic diversion on different segments of highways. Moreover, in the past few decades, more efforts have been invested in developing novel methods and strategies for traffic time prediction with a high requirement on the accuracy and reliability of forecasting. Linear time series analysis has been widely accepted and applied to this research area, such as linear regression models [9], autoregressive integrated moving average (ARIMA) [10, 11] and its extensive extended applications (including spatial-temporal ARIMA [12, 13] and seasonal ARIMA [14, 15]), and Kohonen self-organizing initial classifier [16]. Other studies consider predicting short-term travel time on highways using Bayesian dynamic linear models (DLM) [17], Kalman filtering [18], nonparametric regression models [19], support vector machine models [20, 21], and XGBoost [22], while multiple support vector regression (SVR) models [23] have been used to predict iterative time series. The hidden Markov model (HMM) is performed to forecast short-term traffic during freeway peak periods [24, 25]. Research on traffic prediction by deep learning methods is gradually emerging, and a bilinear recurrent neural network (BLRNN) is used [26], such as LSTM models [27, 28], automatic encoder [29], and CNN-based methods [30, 31]. These methods have abilities to extract features effectively from traffic data.

Furthermore, hybrid models of deep learning are popular because of LSTM units that can find temporal relationships from input sequences and have the ability to extract information features through convolution operations [32–34]. Traffic flow data, which is similar to frequently studied data in the area of machine learning such as video and audio, have plentiful characteristics in both time and space domains. For example: in the space domain, traffic flow patterns sometimes have strong dependencies on nearby locations (topological locality); in the time domain, the present traffic flow will influence the future one. Motivated by the successes of the CNNs and the LSTM and with consideration of the characteristics of travel time, a CNN and an LSTM are combined as the basic frame of the proposed method in this study. Moreover, the attention mechanism is adopted to improve result quality because it can consider differences between input features, which have been proven to be successful in a wide range of tasks [35, 36]. The improved hybrid models that combine CNNs and LSTM can achieve higher prediction accuracy [37, 38]. In addition to directly splicing the CNN module and the LSTM module, the two modules can be merged into one module that is named LSTM-CNN.

In summary, due to the growing demand for real-time travel time information in ITS, a large number of traffic prediction algorithms have been developed. The advantages of hybrid models are gradually highlighted. Hence, we propose an LSTM-CNN strategy to predict the travel time of the target highway with consideration of the correlation of spatial, temporal, and depth components, which can achieve a better performance than the present methods.

3. Methodology

The LSTM-CNN framework for highway travel time forecasting is presented in this section. The method integrates LSTM and CNNs with the attention mechanism and the residual network, so it can have an excellent generalization performance in prediction accuracy and good scalability in highways. As shown in Figure 1, this method mainly includes four steps. In the preliminary prediction step (Step I), the multidimensional inputs of travel-time time series data are grouped by spatial segments to be fed into different LSTM. In the temporal extension step (Step II), the attention mechanism is applied to combine the latest hidden output of LSTM to perform preliminary forecasting. Thus, both spatial and temporal features are stored in the attention-combined output. In the depth extension step (Step III), the residual network is designed to balance the ability to learn the complex spatiotemporal regulation of travel time when nonrecurrent incidents occur and overcome the vanishing gradient problem. In the feature extraction and dimensionality reduction step (Step IV), the complex features from the residual network are extracted with dimensionality reduction by a series of Conv1D layers to achieve the multidimensional final prediction.

3.1. Step I: Preliminary Prediction. A series of travel times of each segment has been input into its LSTM, which composes the spatial input in parallel of LSTM-CNN. The main objective of LSTM is to simulate long-term dependencies of each segment and determine the optimal input length of a gate unit that acts as an input to three multiplying units, blocking or transmitting information based on the importance of the data elements. Estimated weights of data are stored or deleted in a cell by a backpropagating learning process. The mathematical LSTM formulas for each segment are as follows:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + V_i S_{t-1}), \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + V_f S_{t-1}), \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o S_{t-1}), \\ \tilde{S} &= \tanh(W_s x_t + U_s h_{t-1}), \\ S_t &= f_t \times S_{t-1} + i_t \times \tilde{S}, \\ h_t &= o_t \times \tanh(S_t), \end{aligned} \quad (1)$$

where x_t is the input of LSTM from traffic time series at time step t . i_t and o_t are the outputs of the input and the output gates at time step t , respectively. The output of the forgotten

gate, memory cell, and hidden state is represented as f_t , S_t , and h_t , respectively. \tilde{S}_t is the candidate cell to achieve S_t with combination of the memory cell S_{t-1} at time step $t - 1$. W_i , W_f , W_o , and W_s are weights of x_t in the equations of input gate, forgotten gate, output gate, and memory cell, respectively. U_i , U_f , U_o , and U_s are weights of h_{t-1} in the equations of input gate, forgotten gate, output gate, and memory cell, respectively. V_i , V_f , and V_o are weights of S_{t-1} in the equations of input gate, forgotten gate, output gate, and memory cell, respectively. σ is the sigmoid function, which controls the access of information by the output of 0 and 1. $\tanh(\cdot)$ is the hyperbolic tangent function with the range of $[-1.0, 1.0]$, which is the activation function of LSTM. \times is the point multiplication, and $+$ is the addition operation. And the structure diagram of LSTM is shown in Figure 2.

3.2. Step II: Temporal Extension. Although LSTM selects and forgets valuable information from the recurrent time series, there is still a problem in the learning process of LSTM. In LSTM-CNN, spatial information of travel time from different segments are extracted for outputting the travel time of different segments in the next interval, which is a seq2seq model. However, the contribution of the selected information from different segments is dynamic. Methods, such as autoencoder structure, tend to lose information in the extracting process. The attention mechanism is proposed to give dynamic weights for mapping the output of different segments to the final prediction. To consider the state transmission from the downstream to the upstream, S_t is defined as the travel time at any segment at the t interval. The downstream segment is given a smaller serial number, then the n segments, and so on. $S_t, S_{t+1}, \dots, S_{t+n}$ are input into each LSTM and combined by attention mechanism. The softmax activation function is used to deal with α into a weighted vector $\hat{\alpha}$ that belong to $[0, 1]$. Lastly, the weighted vector $\hat{\alpha}$ is applied to combine S_t to acquire the prediction c_j in the next interval. In the weighted vector $\hat{\alpha}$, $\hat{\alpha}_{ij}$ is the spatial information from segment i to the predicted segment j . $\hat{\alpha}$ is the output of the LSTM hidden layer of segment i .

$$c_j = \sum_{i=1}^T \hat{\alpha}_{ij} h_i. \quad (2)$$

3.3. Step III: Depth Extension. The residual network layers (ResNet) are proposed to dig out deep features of spatiotemporal highway segments, which consist of three pairs of 1D convolutional layers and the BatchNorm, followed by a rectified linear unit (ReLU) activation layer. ReLU is used because it can maintain a stable gradient even for larger activations. The next step is to predict future traffic status through the integration of four 1D CNN layers. It combines LSTM with the attention mechanism to grasp more information.

Considering residual network deep processing, the BatchNorm layer has been introduced to accelerate the convergence of the network. With deepening calculations, the network is prone to overfitting and gradient

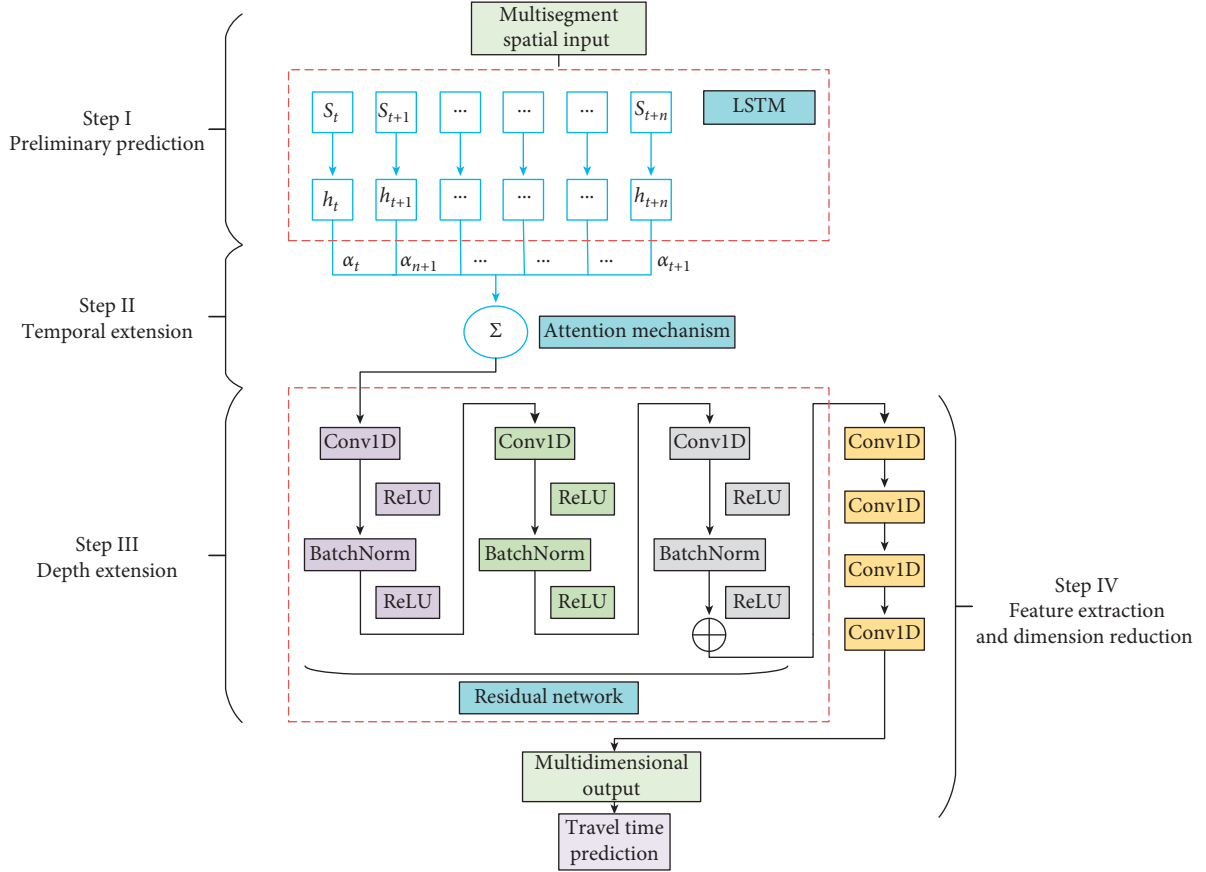


FIGURE 1: LSTM-CNN model structure.

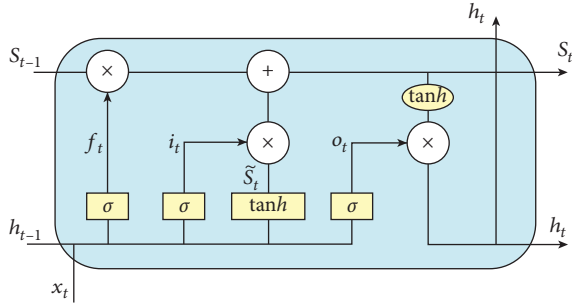


FIGURE 2: LSTM structure diagram.

disappearance. By reducing internal covariance offset, it can quickly converge and avoid overfitting. In deep neural networks, after each single gradient update of a batch of data, each layer will show different feature information than the previous layer. Because parameters of the previous layer are updated during the training, data distribution of the input feature maps is also changed greatly. It also significantly affects the training speed and requires heuristics to determine the initialization of parameters. The BatchNorm layer is a technique that is used to solve the internal covariate offset problem. The normalization of minibatch can be calculated by the following formula:

$$\hat{X}_k = \frac{X_k - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \quad (3)$$

where μ_B and σ_B^2 are mean and variance of the minibatch, respectively; ε is a constant; and X_k and \hat{X}_k are the first and second versions of the k -th input data. Formula (3) can reduce the covariance offset to achieve data standardization. Formula (4) is a learnable parameter, which denotes the scale and rotation values of the input data. The BatchNorm conversion is expressed as follows:

$$Y_k = \gamma \times \hat{X}_k + \beta, \quad (4)$$

where γ and β are learnable parameters and Y_k is a scaling ratio and a rotation value of the k -th input data. In the deep network layer, the parameter initialization is generally close to 0.

When updating the parameters of the shallow network during training, it is easy to cause the gradient to disappear as the network is deepened. Shallow parameters cannot be updated. ResNet is assumed to involve a network layer that is an optimized network level for shallow parameters. Then the designed deep network has many redundant network layers. We hope that these redundant layers can complete the identity mapping to ensure that the input and output pass through the identity layer are exactly the same. The identity

layers are specified and will be self-determined during network training. It can be seen that X is the input of the residual network by this layer. $F(X)$ is the output after the linear change and activation of the first layer. Figure 3 shows the basic framework of the residual network. Before the second layer performs a linear change and activates, $F(X)$ adds the input value X to this layer and then activates the output. X is added before the output value of the second layer. The path is called a shortcut connection. So ResNet meets the exploration of deeper traffic correlations. Without ResNet, the deeper the network is, the harder it is to train with an optimization algorithm. As the depth of the network deepens, training errors will increase.

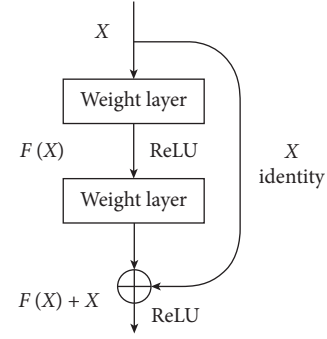


FIGURE 3: Residual network.

3.4. Step IV: Feature Extraction and Dimensionality Reduction. As the output of the residual networks with the fully connected layer is rough, a series of one-dimensional convolution layers is connected to extract valuable features from it to the final prediction. Meanwhile, the one-dimensional convolution layer transmits the dimensionality of the output to the final prediction.

4. Data

4.1. Data Sources. Our data was collected from Shaoxing, Zhejiang Province, China, including data from September 1 to November 30, 2019. The total length of the target highway is about 39.25 km, and the speed limit is 100 km/h. Ascending from the southeast direction to the northwest direction, we divide the entire upward highway into six segments, as shown in Figure 4. Table 1 provides basic information of each segment. There are three toll stations in total, which are Keqiao Toll Station (black circle 1), Shaoxing Toll Station (black circle 2), and 1039 Shangyu Toll Station (black circle 3). There are also four bayonet camera detectors, which are 20311, 20312, 20301, and 20302 from the northwest to the southeast. ANPR cameras were installed in a fixed position to read vehicle license plates with high accuracy. Travel time on roads can be estimated from the data received from the ANPR.

For all 80-day data, we use the first 75-day data as the training set and the last 5-day data as the testing set. LSTM and CNN access the characteristics of traffic status by predicting the statistics of the next 5 min based on the past six 5 min.

Anomalies in the data sets may be much higher or lower than the normal true value, thus providing inaccurate predictions. To suppress the effect of these outliers, we applied the method of winsorization. The purpose of winsorization is to replace the minimum and maximum values in a data set with the closest values. Winsorization plays a crucial role in suppressing the effects of extreme values. Assuming that the value of the sequence to be processed is given by \mathbf{w} , where $\mathbf{w} = (w_1, w_2, \dots, w_c)$, the processed value w_i^C following winsorization will be

$$w_i^C = \begin{cases} w_{i+1}, & \text{if } w_i = \min(\mathbf{w}), \\ w_i, & \text{if } \min(\mathbf{w}) < w_i < \max(\mathbf{w}), \\ w_{c-1}, & \text{if } w_i = \max(\mathbf{w}), \end{cases} \quad (5)$$

where w_i is the i pending value, $\min(\mathbf{w})$ is the minimum value of the pending value, $\max(\mathbf{w})$ is the maximum value of the pending value, c is the number of pending values, and w_i^C is the winsorized value of the i pending value.

4.2. Data Analysis. Frequency distributions are plotted against observed travel times for each of the six segments of the target highway over a two-month period as shown in Figure 5. Each segment shows the distribution of raw data frequencies over the entire data cycle. It is interesting to note that frequencies of segments 2, 3, 5, and 6 are close to Gaussian shapes. The traffic flows in these segments are uninterrupted flows. Segments 1 and 4 did not subject to Gaussian distribution because the traffic flows are interrupted by nearby toll gates in the ramps. Hence, based on the choice of different destinations for different travelers, the results of travel times fluctuate greatly. In particular, there is a Shaoxing service area in the middle of segment 4. Some vehicles may be repaired or under service. The frequency distribution of travel time is further expanded.

Median absolute deviation (MAD) is a parameter for detecting outliers by calculating the sum of distances between observation and average values. The MAD is the average distance between each point and the average point for the overall data fluctuation. MAD is defined as follows:

$$\text{MAD} = \frac{1}{T} \sum_{t=1}^T |\text{tr}_t - \bar{\text{tr}}|. \quad (6)$$

In time domain T , $\bar{\text{tr}}$ is the mean of the average travel time in each interval temporal distribution, and tr_t is the average travel time at interval t . Calculation results of six segments are shown in Table 2. It is clearly seen that the MAD value of segment 4 is much larger than those of other segments, which fully shows that its data are extremely volatile. Among them, segments 2 and 5 are uninterrupted, and the estimated data is very stable. Segment 1 has an exit at

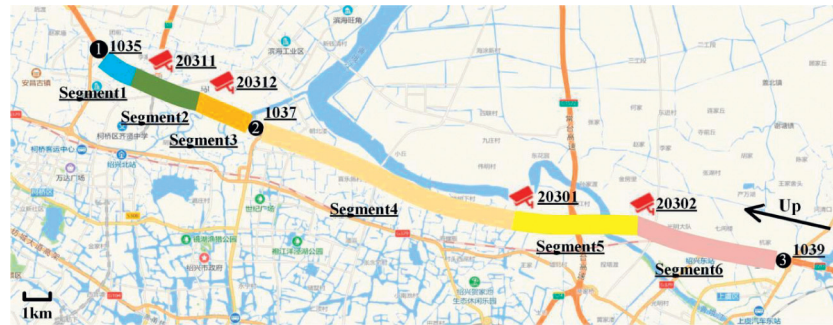


FIGURE 4: Target highway.

TABLE 1: Basic information of each segment.

Highway segment	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6
Length (km)	1.5	4.4	3.5	15.5	6.2	7.6

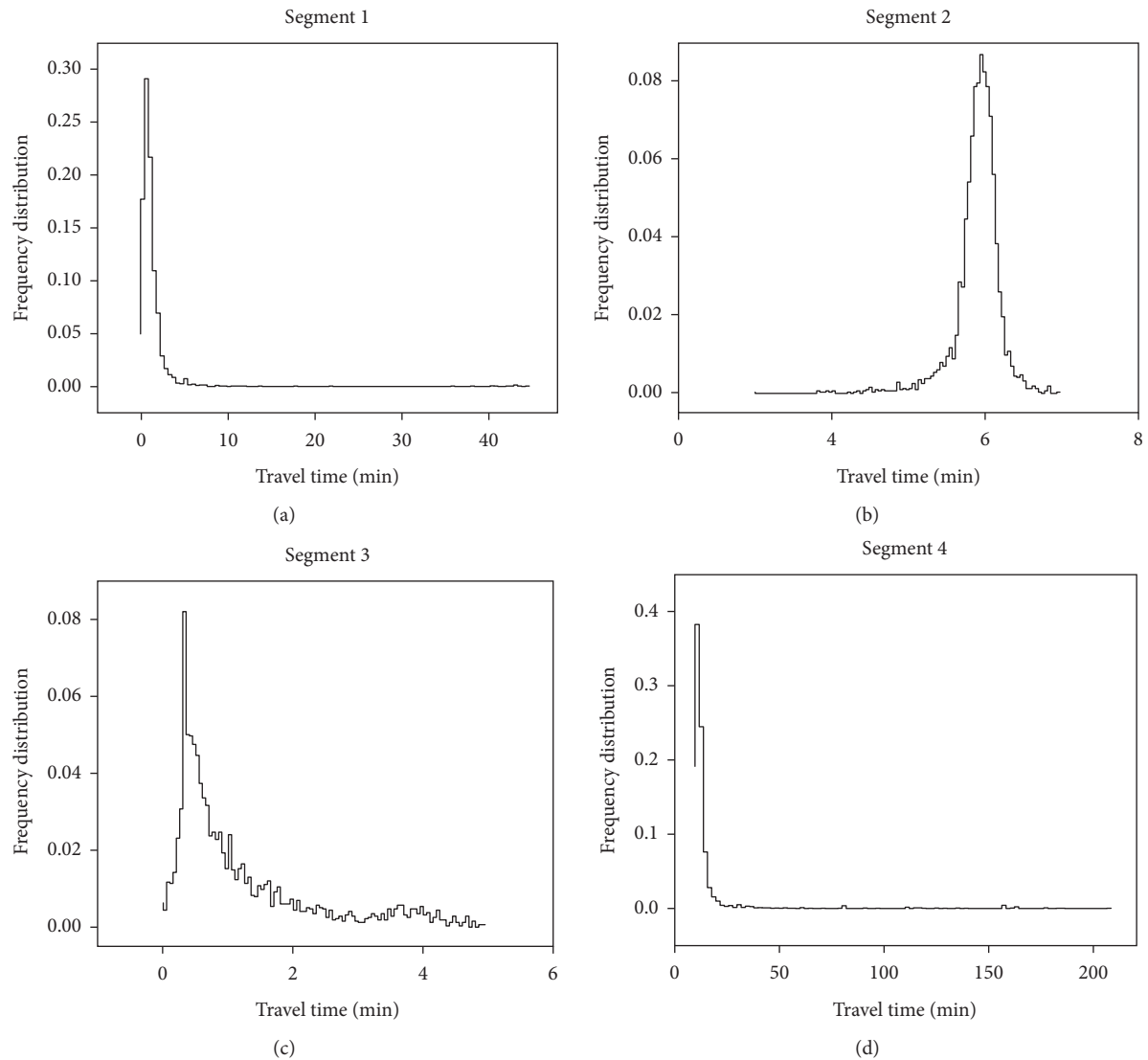


FIGURE 5: Continued.

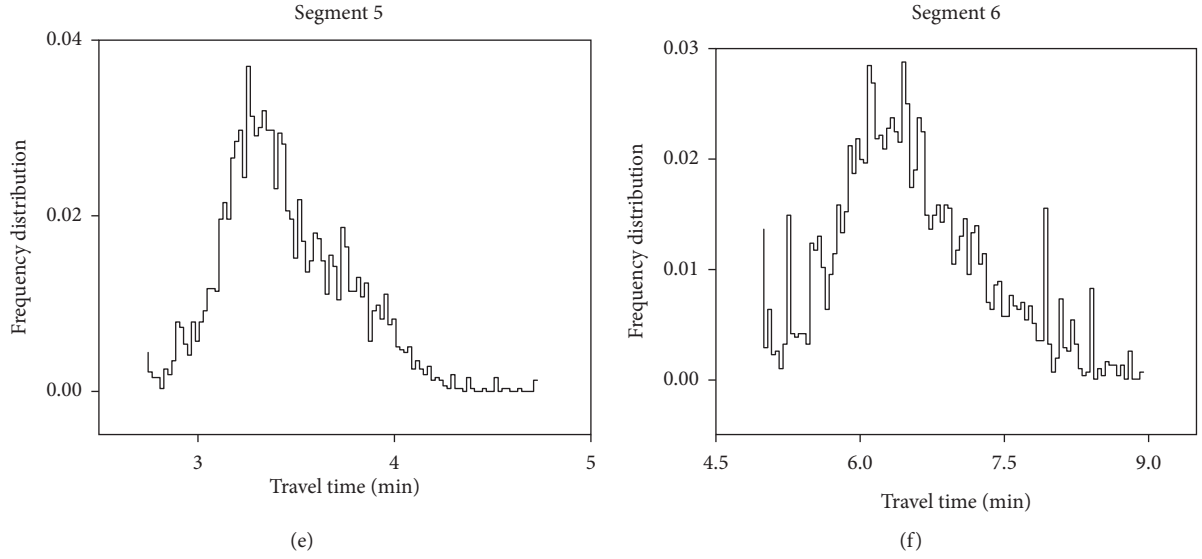


FIGURE 5: Frequency distribution of each segment.

TABLE 2: MAD of each segment.

	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6
MAD	0.4068	0.0785	0.3765	1.4133	0.1874	0.4329

the end and no service area in the segment; therefore, its MAD is smaller than that of segment 4, although it is still much larger than those of segments 2 and 5. For segments 3 and 6, the upstream includes the entrance area of the toll station, and vehicles imported from time to time, which makes data also fluctuate greatly. For different traffic diversions in different segments, we have conducted a study using LSTM-CNN to make predictions. The results demonstrate the accuracy of the proposed model.

5. Case Studies

5.1. Hardware Environment. The hardware environment is as follows: CPU – Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHz, memory – 16 GB, and hard disk – 512 g Samsung EVO850.

5.2. Baseline Methods. The proposed method in this study is compared with the following five methods:

- (1) XGBoost: XGBoost implements a generic tree boosting algorithm. The loss function uses Taylor expansion to the second order, using the first two orders as improved residuals. Regularization is introduced to limit the complexity of the model.
- (2) LR: linear regression is a statistical analysis method that uses regression analysis in mathematical statistics to determine quantitative relationships between two or more interdependent variables.
- (3) SARIMA: In some time series, there are significant cyclical variations. Such cycles are due to seasonal

variations (e.g., quarterly, monthly, and weekly variations) or some other inherent factors. Such series are called seasonal series. The seasonal ARIMA (SARIMA) model is formed by including other seasonal conditions in the ARIMA model. SARIMA is used to deal with the trend and seasonality of the data. In this model, we use one or more seasonal differences to eliminate cyclical variation. SARIMA appears to be more robust than ARIMA.

- (4) CNNs: deep convolutional neural networks are feedforward neural networks that are very similar to ordinary neural networks. They both consist of neurons with learnable weights and bias constants (biases). Each neuron receives some input and does some dot product calculations. The output is a score for each classification.
- (5) LSTM-ST: LSTM-ST is called spatiotemporal LSTM, which is a series of LSTM that is similar to the recurrent structure of Step I (preliminary prediction). The difference is the connection between different LSTM is existing.

5.3. Evaluation Index. Before performing error calculations, we first normalize the test set results. For the evaluation of different prediction methods, we employ root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) as the evaluation indexes. In time domain T , given predicted values \hat{r}_t and ground truth values tr_t , the RMSE, MAE, and MAPE are calculated as follows:

$$\begin{aligned}
\text{RMSE} &= \sqrt{\frac{1}{T} \sum_{t=1}^T (\text{tr}_t - \hat{\text{tr}}_t)^2}, \\
\text{MAE} &= \frac{1}{T} \sum_{t=1}^T |(\text{tr}_t - \hat{\text{tr}}_t)|, \\
\text{MAPE} &= \sum_{t=1}^T \left| \frac{\text{tr}_t - \hat{\text{tr}}_t}{\text{tr}_t} \right| \times \frac{100\%}{T}.
\end{aligned} \tag{7}$$

5.4. Results. The accuracy of the proposed predicting method is examined based on the traffic levels and time of the day for each segment. Figure 6 shows the performance of traffic travel time forecasting using the proposed method, that is, MAE, MAPE, and RMSE by variation in segment levels and time of day. The box plot shows the distribution of prediction errors, with the solid red line representing the mean of the errors. It is observed that travel time varies from one segment to another over the forecast time. It can be seen from different fluctuations in MAPE. Segment 4 has the largest error, which is followed by segments 6 and 1. Similarly, when examining the nature of the errors corresponding to the time of day, the prediction accuracy is measured with a peak of 7:00 to 8:00. The error is higher during the peak hour. It is relatively low in the off-peak hours. Overall, the distribution in Figure 6 shows that travel times in different segments are affected differently at different times of a day. In all, the results show that the LSTM-CNN model provides reliable and accurate predictions of travel time.

As shown in Figure 6, the results of segment 4 fluctuate most significantly due to the various travel behaviors of different drivers. Figure 7 shows the forecast results from segment 4 for October 16 to October 21. The purple shaded area shows that when there is a sharp increase of travel time, the ground true curve (red) is very similar to the model prediction curve (green). In this case, the approach presented in this paper helps obtain the desired performance. The predictions of segment 6 have a flatter fluctuation, as shown in Figure 8, while the predictions of segment 2 have the least fluctuation in the data, as shown in Figure 9. It can be seen that for each different traffic diversion case, the predictive accuracy of the LSTM-CNN model is excellent.

Figure 10 shows that the error results of six segments at different time steps, indicating that the model is better at predicting short travel times. The predicted accuracy for the next six steps of different methods is given in Tables 3–5. LSTM-CNN shows the best performance in multistep prediction than other benchmarks with both spatial and temporal learning ability.

5.5. Comparison. Figure 11 shows prediction error results of LSTM-CNN, LSTM-ST, and CNNs in 5 days of the mean errors that are indicated by different heights. The results show that there is a statistically significant difference in the mean

prediction error between the other two methods and the LSTM-CNN method. As shown in Table 6, as for the RMSE from days 1 to 5, LSTM-CNN method outperforms CNN method by 59.18%, 45.22%, 45.25%, 54.02%, and 72.37%, respectively, and LSTM-ST method by 6.77%, 32.06%, 1.39%, 2.21%, and 29.51%, respectively. In the MAE, LSTM-CNN method outperformed CNN method by 66.93%, 56.49%, 58.65%, 67.15%, and 75.60%, respectively, and LSTM-ST method by 15.00%, 41.87%, 15.76%, 21.33%, and 36.83%, respectively. For the MAPE, LSTM-CNN method outperformed CNN method by 66.75%, 55.92%, 58.31%, 66.89%, and 75.58%, respectively, and LSTM-ST method by 14.51%, 41.38%, 15.59%, 21.02%, and 36.47%, respectively.

For the visualization of one-dimensional data, both histogram and kernel density estimates are good ways to represent the probability distribution of individual data values, but the two methods for representing the cumulative distribution of data are helpless. Cumulative distribution of the data, which is the probability distribution of all data less than or equal to the current data value, is not useful for indicating. Data points are in an interval of the probability of occurrence. Mathematically speaking, cumulative distribution function (CDF) is the integral of the probability distribution function. And when plotting CDF, the true probability distribution function is unknown. Therefore, it is often defined as the integral of histogram distribution, as shown in the following formula:

$$\text{cdf}(x) \approx \int_{-\infty}^x dt \text{histo}(t). \tag{8}$$

Figure 12 plots the accumulation of RMSE for LSTM-CNN, CNNs, and LSTM-ST of the distribution function. They demonstrate the statistical properties of the three methods separately. Experimental results demonstrate the validity of the proposed method for travel time prediction.

Figure 13 shows the mean errors of CNNs, LR, LSTM-CNN, LSTM-ST, SARIMA, and XGBoost, with the left axis showing the three errors of the box and the right axis showing the mean line. Table 7 shows specific error values for different methods. LSTM-CNN has the best performance for all three types of errors in six segments.

Short-term predictions are mainly used for short-range trip planning and are desired by travelers. For this purpose, based on historical data, we predict the following (5 min, 10 min, and 15 min) trips time. Figure 14 and Table 8 list results of XGBoost, LR, SARIMA, CNNs, LSTM-ST, and LSTM-CNN.

We compare LSTM-CNN with the five other methods. It is clearly observed that LSTM-CNN produces the most accurate short-term traffic travel time prediction considering these three errors. In the mean results of short-term prediction shown in Table 8, the RMSE of LSTM-CNN was better than XGBoost, LR, SARIMA, CNNs, and LSTM-ST method by 20.41%, 18.93%, 13.04%, 22.27%, and 30.72%, respectively. As for the MAE, the improvements were 29.60%, 46.27%, 25.90%, 59.58%, and 49.81%, respectively, whereas for the MAPE, the improvements were 19.41%, 49.40%, 30.40%, 62.49%, and 44.92%, respectively. LSTM-ST method exhibited the worst predictive performance.

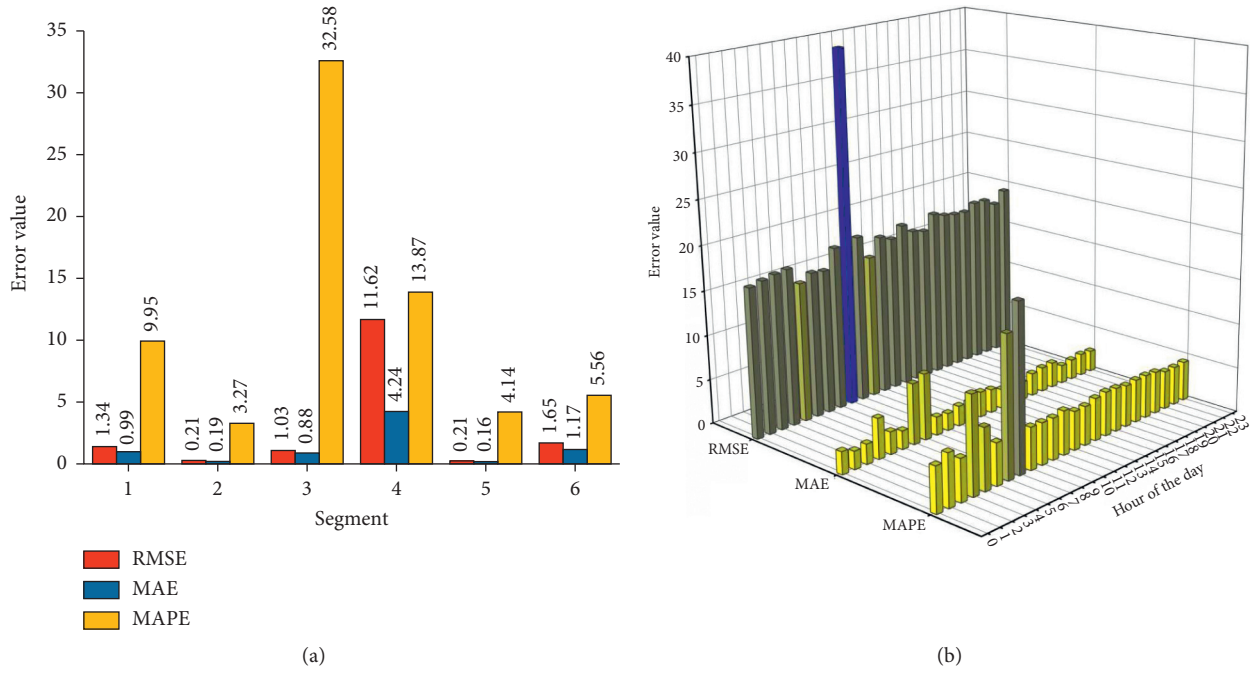


FIGURE 6: Forecast errors for LSTM-CNN by different segments and hours of the day.

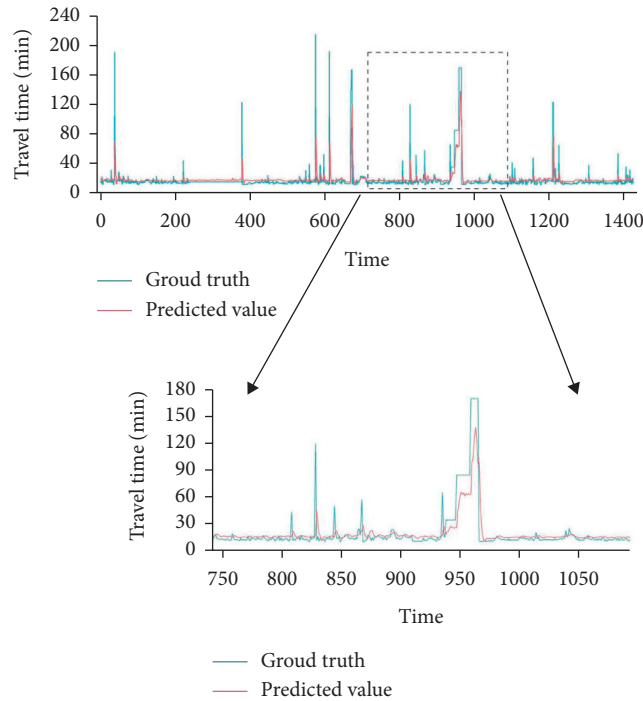


FIGURE 7: Prediction result of segment 4.

Long-term prediction is used primarily by long-distance travelers who plan their trips in advance, and it is considered more challenging than short-term prediction. The setup predicted the next 20 min, 40 min, and 60 min trip times based on historical data. Figure 15 and Table 9 list results of XGBoost, LR, SARIMA, CNNs, LSTM-ST, and LSTM-CNN.

Similarly, we compare LSTM-CNN with five other methods. It is clearly observed that LSTM-CNN produces the most accurate long-term traffic trip time predictions for the three errors. According to results in Table 9, the RMSE of LSTM-CNN was lower than XGBoost, LR, SARIMA, CNNs, and LSTM-ST methods by 7.76%, 13.67%, 6.51%, 6.28%, and

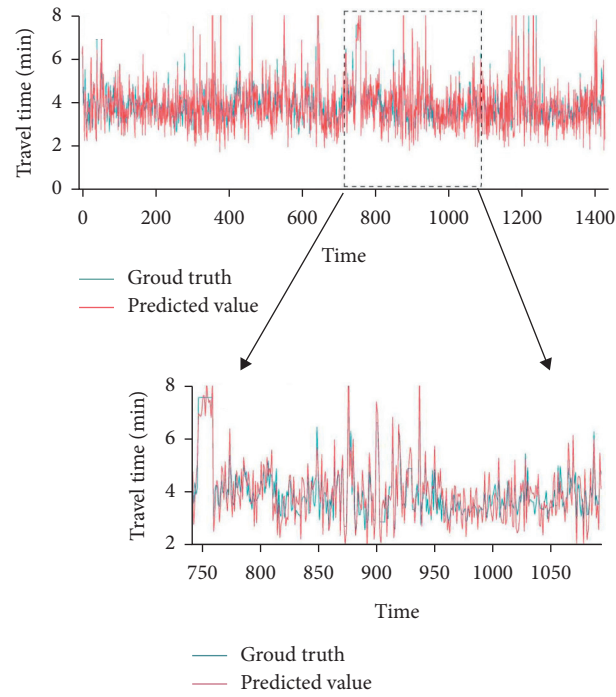


FIGURE 8: Prediction result of segment 6.

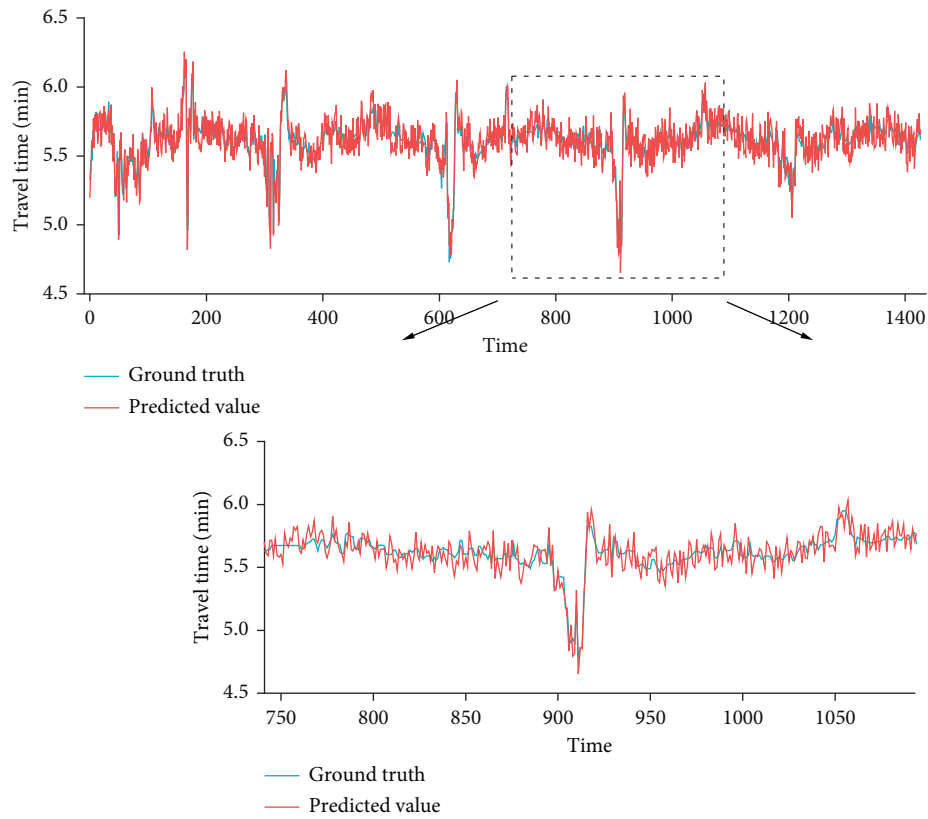


FIGURE 9: Prediction result of segment 2.

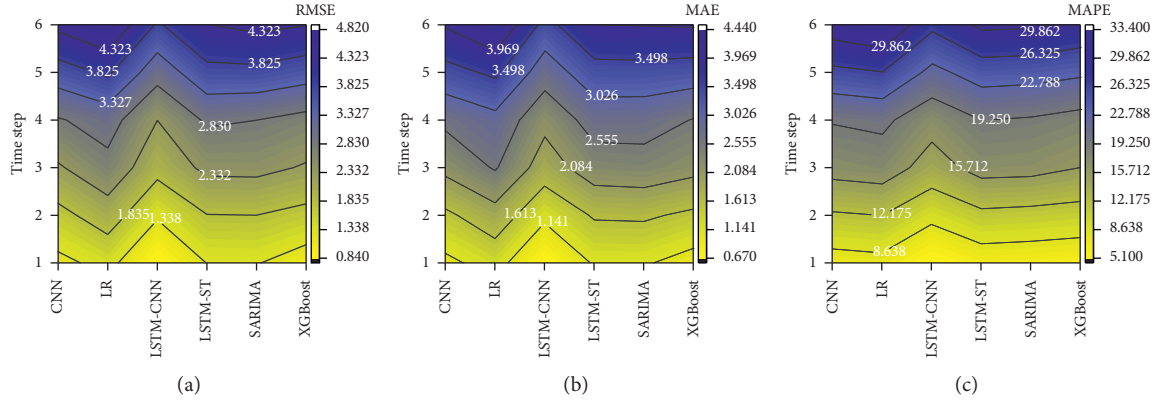


FIGURE 10: Error results of different time steps.

TABLE 3: RMSE comparison of multistep prediction.

RMSE	1	2	3	4	5	6
CNNs	1.2292	1.6882	2.2781	2.7891	3.5901	4.4493
LR	1.4719	2.0813	2.6873	3.0333	3.9102	4.8009
LSTM-CNN	0.8400	1.3886	1.9845	2.3349	3.0157	3.7664
LSTM-ST	1.3467	1.8231	2.4452	2.8992	3.6882	4.3112
SARIMA	1.3598	1.8334	2.4564	2.8312	3.7005	4.4481
XGBoost	1.1189	1.6912	2.2903	2.6773	3.5421	4.3337

TABLE 4: MAE comparison of multistep prediction.

MAE	1	2	3	4	5	6
CNNs	1.0471	1.5187	2.2118	2.6556	3.3382	4.0167
LR	1.3098	1.9034	2.598	2.8892	3.5901	4.4312
LSTM-CNN	0.6736	1.2296	1.8536	2.2079	2.7699	3.3381
LSTM-ST	1.1811	1.6611	2.3356	2.7512	3.3198	3.9672
SARIMA	1.1826	1.6782	2.3781	2.7375	3.3285	3.9883
XGBoost	0.9745	1.5219	2.2209	2.5353	3.2665	4.0102

TABLE 5: MAPE comparison of multistep prediction.

MAPE (%)	1	2	3	4	5	6
CNNs	7.3242	11.7765	16.9902	19.4753	25.4663	31.8992
LR	7.6883	12.1982	17.5472	19.9845	26.2481	33.3356
LSTM-CNN	5.1379	9.4568	14.2378	16.9844	21.8500	27.1055
LSTM-ST	6.7568	11.4241	16.923	19.2761	24.3654	30.5805
SARIMA	6.5987	11.1123	16.7732	18.9345	24.1008	30.3562
XGBoost	6.2561	10.7331	15.6931	18.1032	23.3214	29.1004

8.76%, respectively. Its MAE was better than others by 3.05%, 39.39%, 15.02%, 27.28%, and 22.38%, respectively. Its MAPE was better than others by 8.14%, 46.20%, 22.42%, 32.77%, and 22.64%, respectively. The MAPE of XGBoost was better than that of LSTM-CNN in the long-range

prediction. Based on the percentage deviation from observations, MAPE can better predict the accuracy. Also, we see that as the length of the prediction increased, the gap among methods became smaller. The prediction performance of all methods decreased significantly.

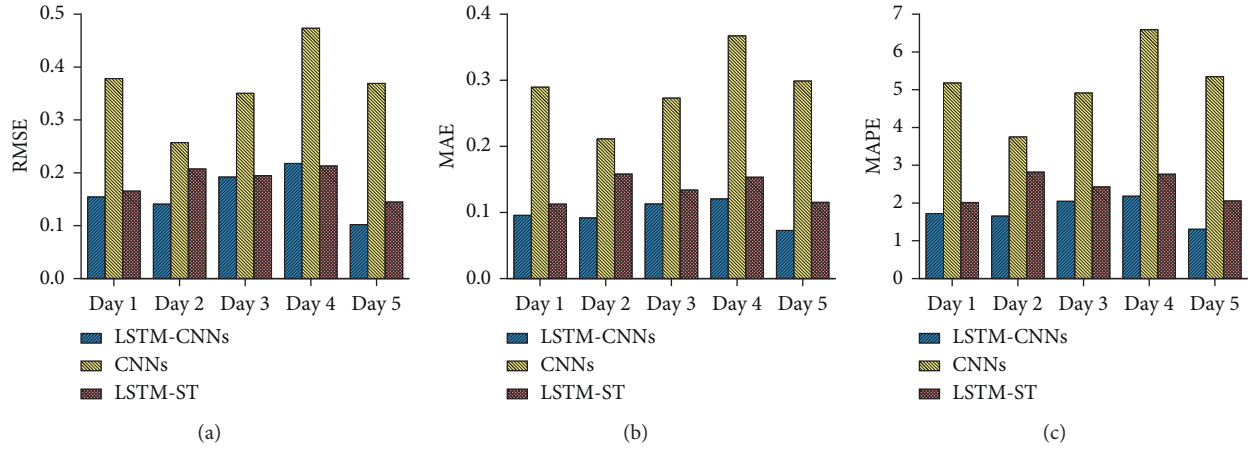


FIGURE 11: Comparison of results for LSTM-CNN, CNN, and LSTM-ST.

TABLE 6: Error results of LSTM-CNN, CNN, and LSTM-ST for 5 days.

Error	Methods	Day 1	Day 2	Day 3	Day 4	Day 5
RMSE	LSTM-CNN	0.1543	0.1409	0.1919	0.2177	0.1020
	CNNs	0.3780	0.2572	0.3505	0.4735	0.3691
	LSTM-ST	0.1655	0.2074	0.1946	0.2130	0.1447
MAE	LSTM-CNN	0.0958	0.0919	0.1128	0.1206	0.0729
	CNNs	0.2897	0.2112	0.2730	0.3671	0.2988
	LSTM-ST	0.1127	0.1581	0.1339	0.1533	0.1154
MAPE (%)	LSTM-CNN	1.7222	1.6533	2.0477	2.1821	1.3048
	CNNs	5.1798	3.7511	4.9118	6.5902	5.3440
	LSTM-ST	2.0145	2.8202	2.4260	2.7627	2.0538

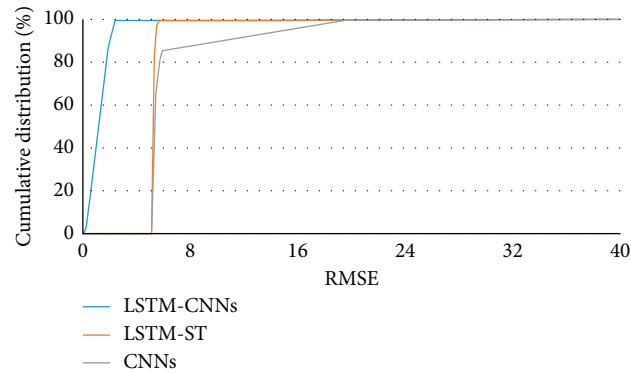


FIGURE 12: CDF results.

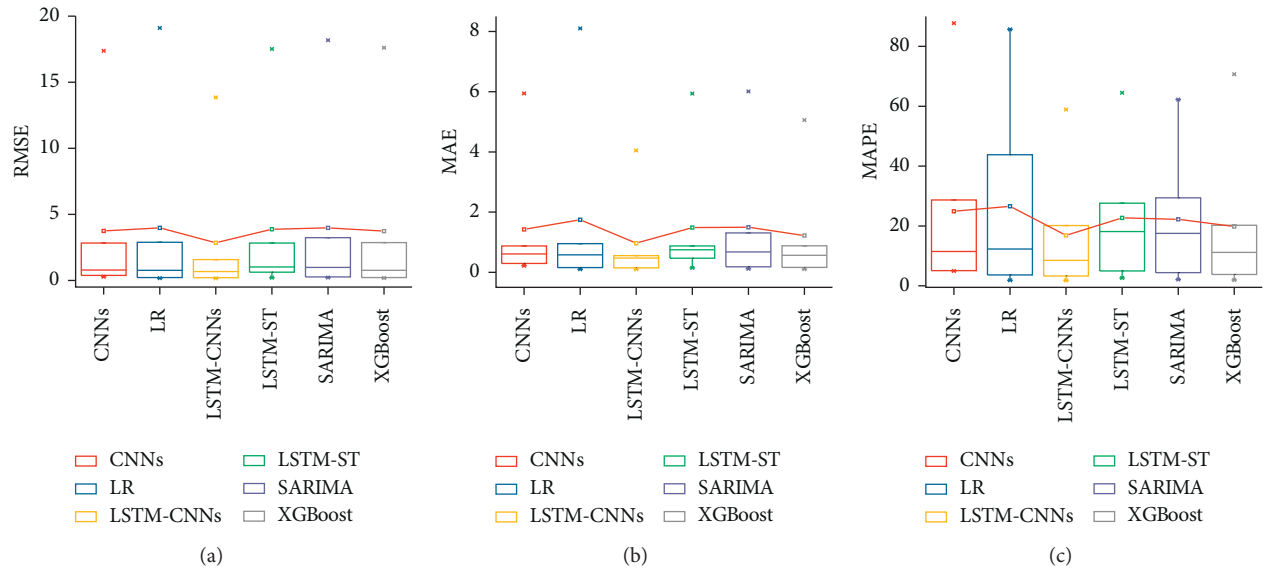


FIGURE 13: Error results of different methods.

TABLE 7: Error results of different methods.

Methods	Error	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6
CNNs	RMSE	2.8242	0.3846	0.8826	17.3829	0.2771	0.6939
	MAE	0.8733	0.2945	0.6959	5.9437	0.2194	0.526
	MAPE (%)	17.7045	5.2918	87.761	28.6955	4.9691	5.087
LR	RMSE	2.8847	0.1672	0.8917	19.1103	0.2163	0.6313
	MAE	0.9538	0.1031	0.6882	8.1038	0.159	0.4745
	MAPE (%)	19.5272	1.8677	85.7107	43.7899	3.6489	5.1051
LSTM-CNN	RMSE	1.5766	0.1666	0.7718	13.8563	0.2097	0.5642
	MAE	0.5549	0.0989	0.5418	4.0471	0.1452	0.4054
	MAPE (%)	12.8176	1.7848	58.9024	20.1201	3.305	4.2412
LSTM-ST	RMSE	2.8254	0.2065	1.1961	17.5173	0.8503	0.6259
	MAE	0.8774	0.1446	0.8220	5.9398	0.6748	0.4702
	MAPE (%)	17.8440	2.6093	64.5140	27.6214	18.4803	4.9916
SARIMA	RMSE	3.2309	0.2162	1.1533	18.1823	0.2649	0.8145
	MAE	1.3102	0.1183	0.7627	6.0104	0.1846	0.5878
	MAPE (%)	27.2927	2.1366	62.2503	29.4217	4.4407	7.7525
XGBoost	RMSE	2.8687	0.1823	0.8844	17.6053	0.2217	0.6523
	MAE	0.8792	0.1082	0.6361	5.0568	0.1633	0.4885
	MAPE (%)	17.3071	1.9531	70.682	20.2381	3.8021	5.0538

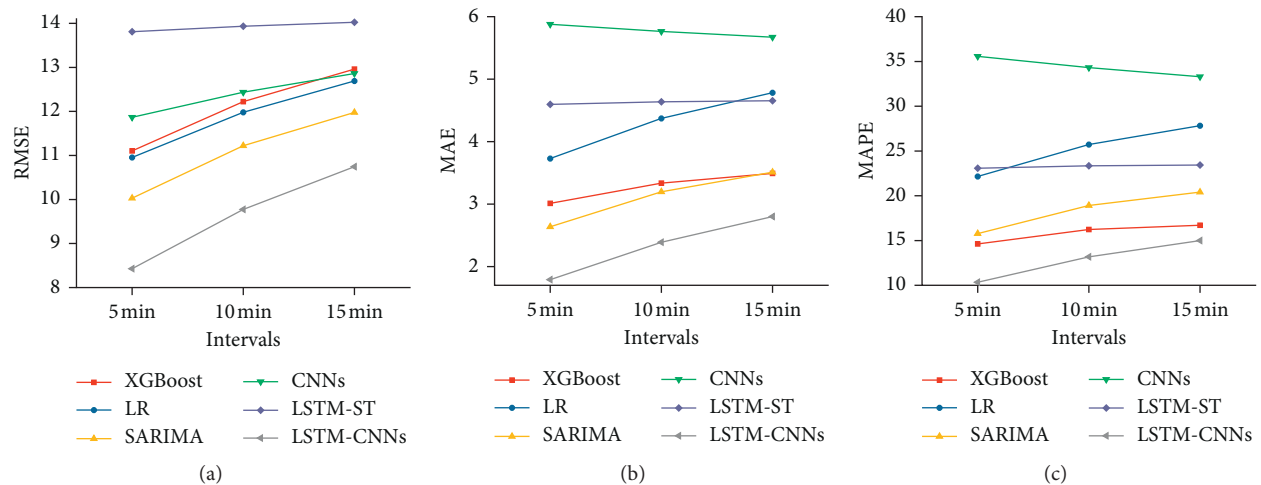


FIGURE 14: Short-term prediction error results.

TABLE 8: Short-term prediction error results of different methods.

Methods	Error	5 min	10 min	15 min
XGBoost	RMSE	11.1040	12.2234	12.9625
	MAE	3.0112	3.3359	3.4920
	MAPE (%)	14.6106	16.2274	16.7007
LR	RMSE	10.9548	11.9806	12.6909
	MAE	3.7284	4.3713	4.7821
	MAPE (%)	22.1522	25.7278	27.8272
SARIMA	RMSE	10.0307	11.2200	11.9780
	MAE	2.6366	3.1975	3.5113
	MAPE (%)	15.7659	18.9068	20.3997
CNNs	RMSE	11.8657	12.4348	12.8615
	MAE	5.8780	5.7638	5.6719
	MAPE (%)	35.5655	34.3175	33.2884
LSTM-ST	RMSE	13.8093	13.9326	14.0234
	MAE	4.5960	4.6382	4.6554
	MAPE (%)	23.0800	23.3385	23.4371
LSTM-CNN	RMSE	8.4288	9.7757	10.7455
	MAE	1.7900	2.3881	2.7997
	MAPE (%)	10.3360	13.1795	14.9992

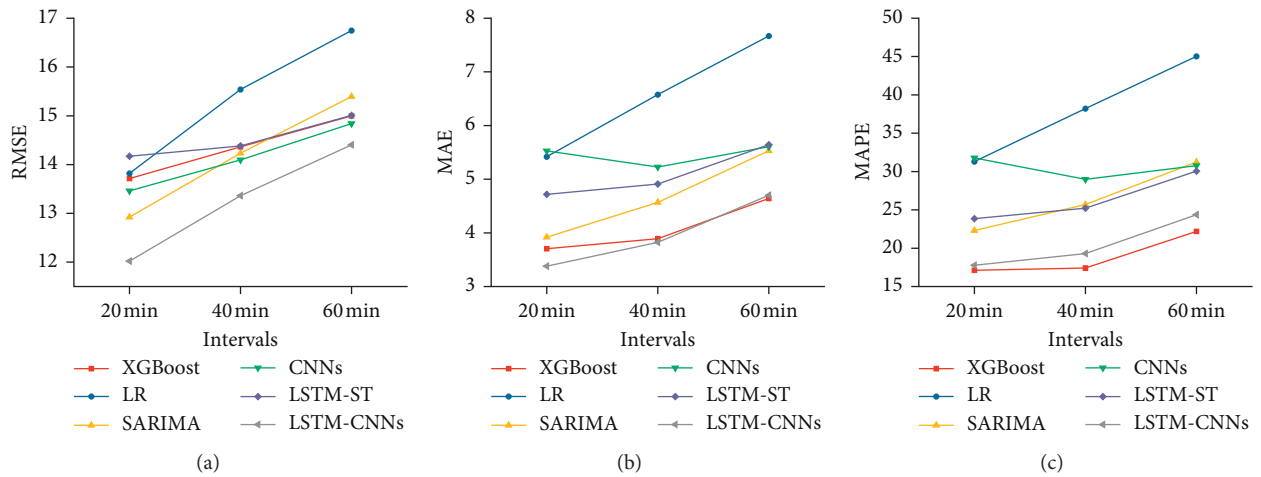


FIGURE 15: Long-term prediction error results.

TABLE 9: Long-term prediction error results of different methods.

Methods	Error	20 min	40 min	60 min
XGBoost	RMSE	13.7142	14.3634	14.9989
	MAE	3.7057	3.8906	4.6398
	MAPE (%)	17.1187	17.4168	22.1957
LR	RMSE	13.8199	15.5390	16.7464
	MAE	5.4188	6.5753	7.6697
	MAPE (%)	31.2934	38.2185	45.0381
SARIMA	RMSE	12.9227	14.2330	15.3928
	MAE	3.9210	4.5686	5.5299
	MAPE (%)	22.3183	25.6994	31.2392
CNNs	RMSE	13.4614	14.0973	14.8385
	MAE	5.5265	5.2283	5.6061
	MAPE (%)	31.7741	28.9882	30.7723

TABLE 9: Continued.

Methods	Error	20 min	40 min	60 min
LSTM-ST	RMSE	14.1691	14.3815	15.0072
	MAE	4.7177	4.9093	5.6427
	MAPE (%)	23.8613	25.2212	30.0549
LSTM-CNN	RMSE	12.0201	13.3627	14.4035
	MAE	3.3805	3.8236	4.7018
	MAPE (%)	17.7759	19.3002	24.3612

6. Conclusion

The ability to predict highway travel time in a timely and accurate manner is essential for proactive traffic management strategies that helps provide reliable services to travelers. In this paper, we build upon residual network and attention mechanism and propose a prediction method for the combined LSTM-CNN method. The collected and processed high-speed uplink data sets are predicted segment by segment. We consider traffic diversion to evaluate multisteps performance and test the method in a real-world case to predict 5-day travel time. RMSE, MAE, and MAPE are used as evaluation metrics. The results show that the proposed method is superior to XGBoost, SARIMA, CNNs, LR, and LSTM-ST methods, indicating that it can contribute to the improvement of the traffic travel time prediction.

Data Availability

The data that support the findings of this study were provided by Zhejiang Communications Investment Group Co., Ltd. Restrictions apply to the availability of these data, which were used under license for this study.

Disclosure

The views present in this paper are those of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the “Research on Collaborative Information Interaction Technology between Roads and Vehicles of Intelligent Highway and Its Application” sponsored by the Department of Transportation of Zhejiang Province (2020006) and the National Key Research and Development Program of China (2018YFB1601000).

References

- [1] S. K. Shanmuganathan, “A HMM-based prediction model for spatio-temporal trajectories,” M. S. thesis, The University of Texas at Arlington, Arlington, TX, USA, 2014.
- [2] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “DNN-based prediction model for spatio-temporal data,” in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4, Burlingame CA, USA, October 2016.
- [3] H. Shao and B. H. Soong, “Traffic flow prediction with long short-term memory networks (LSTMs),” in *Proceedings of the 2016 IEEE Region 10 Conference (TENCON)*, pp. 2986–2989, IEEE, Singapore, November 2016.
- [4] Y. Liu, H. Zheng, X. Feng, and Z. Chen, “Short-term traffic flow prediction with Conv-LSTM,” in *Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, IEEE, Nanjing, China, October 2017.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [6] S. Wu, S. Zhong, and Y. Liu, “Deep residual learning for image steganalysis,” *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10437–10453, 2018.
- [7] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, Long Beach, CA, USA, December 2017.
- [8] B. Sun, T. Sun, Y. Zhang, and P. Jiao, “Urban traffic flow online prediction based on multi-component attention mechanism,” *IET Intelligent Transport Systems*, vol. 14, no. 10, pp. 1249–1258, 2020.
- [9] J. Patnaik, S. Chien, and A. Bladikas, “Estimation of bus arrival times using APC data,” *Journal of Public Transportation*, vol. 7, no. 1, pp. 1–20, 2004.
- [10] M. Levin and Y. D. Tsao, “On forecasting freeway occupancies and volumes (abridgment),” *Transportation Research Record*, no. 773, pp. 47–49, 1980.
- [11] H. Q. Huang and T. H. Tang, “Short-term traffic flow forecasting based on ARIMA-ANN,” in *2007 IEEE International Conference on Control and Automation*, pp. 2370–2373, Seoul, Republic of Korea, October 2007.
- [12] R. K. Pace, R. Barry, J. M. Clapp, and M. Rodriguez, “Spatiotemporal autoregressive models of neighborhood effects,” *The Journal of Real Estate Finance and Economics*, vol. 17, no. 1, pp. 15–33, 1998.
- [13] P. Duan, G. Mao, C. Zhang, and S. Wang, “STARIMA-based traffic prediction with time-varying lags,” in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems*, pp. 1610–1615, Rio de Janeiro, Brazil, November 2016.
- [14] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal ARIMA model with limited input data,” *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.
- [15] A. Guin, “Travel time prediction using a seasonal autoregressive integrated moving average time series model,” in *Proceedings of the 2006 IEEE International Conference on Intelligent Transportation Systems*, pp. 493–498, Toronto, ON, Canada, September 2006.

- [16] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.
- [17] X. Fei, C.-C. Lu, and K. Liu, "A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1306–1318, 2011.
- [18] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.
- [19] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Non-parametric estimation of route travel time distributions from low-frequency floating car data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 343–362, 2015.
- [20] C. H. Zheng, L. C. Jiao, and A. L. Ding, "Automatic model selection for support vector machines using heuristic genetic algorithm," *Control Theory & Applications*, vol. 23, no. 2, pp. 187–192, 2006.
- [21] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, "Iterated time series prediction with multiple support vector regression models," *Neurocomputing*, vol. 99, pp. 411–422, 2013.
- [22] B. Sun, T. Sun, and P. Jiao, "Spatio-temporal segmented traffic flow prediction with ANPRS data based on improved XGBoost," *Journal of Advanced Transportation*, vol. 2021, Article ID 5559562, 2021.
- [23] J. N. Wang, X. M. Chen, and S. X. Guo, "Bus travel time prediction model with ν -support vector regression," in *Proceedings of the 12th International Conference on Intelligent Transportation Systems*, pp. 1–6, St. Louis Missouri, MO, USA, April 2009.
- [24] Y. Qi and S. Ishak, "Application of hidden Markov models to short-term speed prediction during peak periods," in *Proceedings of the Transportation Research Board Meeting*, Washington DC, USA, January 2011.
- [25] Y. Qi and S. Ishak, "A Hidden Markov Model for short term prediction of traffic conditions on freeways," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 95–111, 2014.
- [26] D.-C. Park, "Structure optimization of BiLinear recurrent neural networks and its application to ethernet network traffic prediction," *Information Sciences*, vol. 237, pp. 18–28, 2013.
- [27] P. He, G. Jiang, S.-K. Lam, and D. Tang, "Travel-time prediction of bus journey with multiple bus trips," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4192–4205, 2019.
- [28] W. Xiangxue, X. Lunhui, and C. Kaixun, "Data-driven short-term forecasting for urban road network traffic based on data processing and LSTM-RNN," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3043–3060, 2019.
- [29] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: a deep learning approach[J]," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [30] C. Song, H. Lee, C. Kang, W. Lee, Y. B. Kim, and S. W. Cha, "Traffic speed prediction under weekday using convolutional neural networks concepts," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1293–1298, Los Angeles, CA, USA, June 2017.
- [31] X. Ran, Z. Shan, Y. Fang, and C. Lin, "A convolution component-based method with attention mechanism for travel-time prediction," *Sensors*, vol. 19, no. 9, 2019.
- [32] Z. Xu, S. Li, and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in *Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition*, pp. 141–145, Kuala Lumpur, Malaysia, November 2015.
- [33] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, "Improved deep hybrid networks for urban traffic flow prediction using trajectory data," *IEEE Access*, vol. 6, pp. 31820–31827, 2018.
- [34] R. Cai, B. Zhu, L. Ji, T. Hao, J. Yan, and W. Liu, "An CNN-LSTM attention approach to understanding user query intent from online health communities," in *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 430–437, New Orleans, LA, USA, November 2017.
- [35] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Computation and Language*, 2015, <https://arxiv.org/abs/1508.04025>.
- [36] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3459–3471, 2018.
- [37] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional Bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.
- [38] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: a deep learning approach," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5044–5059, 2018.