Proceedings of CONF-MLA 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA36282

Comparative Analysis of LSTM and PatchTST Models for Urban Street Traffic Time Series Forecasting

Wentao Hu[1*], Er Zhou[2]

[1]*College of Artificial Intelligence, Chongqing Technology and Business University, Chongqing, China*
[2]*Faculty of Science, The University of British Columbia, Vancouver, Canada*
**Corresponding Author. Email: huwentao@ctbu.edu.cn*

Abstract. With the increasing complexity of urban traffic systems, reliable forecasting of road conditions has become an important requirement for congestion management and intelligent transportation applications. This study develops a multivariate and multi-horizon traffic speed forecasting framework using real-world traffic monitoring data collected from 75 street segments in Shenzhen, China. LSTM and PatchTST are selected as representative recurrent and Transformer-based forecasting models for comparative evaluation. Average travel speed is considered the prediction target, while traffic index, total sample travel length, and total sample travel time are incorporated as additional input variables. Historical observations from the previous 48 hours are used to forecast traffic speeds over three future horizons, including 30, 60, and 120 minutes. To increase the dependability of experimental outcomes, the dataset is split chronologically, and several random seeds are used. MAE, RMSE, MAPE, and R2 are used to assess the model's performance. According to the experimental findings, PatchTST consistently outperforms LSTM in predicting across all prediction horizons. The MAE values of PatchTST are 2.0710, 2.1960, and 2.3552 for 30-, 60-, and 120-minute forecasting tasks, corresponding to reductions of 6.81%, 7.86%, and 5.89% compared with LSTM. Meanwhile, PatchTST obtains R^2 values of 0.8886, 0.8795, and 0.8670, respectively. Although the forecasting accuracy of both models decreases with longer prediction horizons, PatchTST maintains relatively lower errors and better stability. The results indicate that the patch-based representation and attention mechanism provide advantages in capturing both local variations and long-term temporal dependencies in urban traffic sequences.

Keywords: Traffic speed prediction, LSTM, PatchTST, time series prediction, intelligent transportation

1. Introduction

Urban traffic systems have become increasingly dynamic due to continuous growth in travel demand and the complexity of road operating environments. Accurate forecasting of future traffic conditions plays an essential role in congestion mitigation, traffic management, and intelligent transportation services. Short-term traffic forecasting aims to extract useful temporal patterns from historical

DOI: 10.54254/2755-2721/2026.BA36282

observations and provide reliable predictions for future traffic states. However, traffic sequences are usually influenced by multiple factors, including variations in travel behavior, road conditions, and unexpected events, resulting in strong nonlinearity and uncertainty [1].

The rapid development of traffic sensing and data collection technologies has provided large-scale and high-frequency traffic records, creating favorable conditions for data-driven forecasting methods. Compared with traditional statistical approaches, deep learning models are capable of automatically learning complex representations from historical data and have demonstrated strong performance in nonlinear time series prediction problems. Lv et al. explored deep learning-based traffic prediction methods and verified their effectiveness in handling large-scale traffic datasets [2].

Among existing deep learning approaches, Long Short-Term Memory networks (LSTM) have been widely applied in traffic forecasting because of their ability to model sequential dependencies through recurrent structures [3]. Ma et al investigated LSTM-based traffic speed prediction and demonstrated the potential of recurrent neural networks for capturing temporal variations in traffic conditions [4]. Furthermore, attention-enhanced LSTM models have been introduced to improve the extraction of important temporal information from historical sequences [5]. However, due to its sequential processing mechanism, LSTM may encounter difficulties when modeling very long historical sequences and capturing distant temporal relationships.

Recently, Transformer-based architectures have attracted increasing attention in time series forecasting because of their capability to establish global dependencies through self-attention mechanisms. Transformer models provide an alternative solution for long sequence modeling by directly learning relationships among different temporal positions [6]. Based on this idea, PatchTST introduces a patch-level representation strategy, where continuous observations are grouped into local segments and processed through Transformer encoders to improve long-term forecasting ability [7]. In addition, recent studies such as TimesNet and iTransformer have further explored the potential of Transformer architectures for general time series forecasting tasks [8, 9].

Although deep learning methods have achieved considerable progress in traffic forecasting, several issues remain in practical urban street scenarios. First, many existing studies focus on individual traffic variables, while real-world traffic states are usually affected by multiple correlated factors. Therefore, the effectiveness of multivariate information integration requires further investigation. Second, although LSTM and Transformer-based models have shown promising forecasting capabilities, their performance differences under identical datasets, input conditions, and forecasting horizons have not been fully analyzed.

To address these issues, this study constructs a multivariate traffic speed forecasting task based on real-time street traffic data collected in Shenzhen, China. LSTM and PatchTST are compared under the same experimental framework, with average travel speed selected as the forecasting target and traffic index, total sample travel length, and total sample travel time used as auxiliary variables. Three forecasting horizons, including 30, 60, and 120 minutes, are considered. MAE, RMSE, MAPE, and R^2 are adopted as evaluation metrics to investigate the applicability and performance differences between recurrent and Transformer-based forecasting models in urban street traffic scenarios.

2. Data and methods

2.1. Data sources and prediction task construction

This study uses real-time traffic monitoring data of urban streets released by the Shenzhen Open Data Platform for research [10]. The dataset contains traffic operation data of some streets collected

DOI: 10.54254/2755-2721/2026.BA36282

from March 2018 to November 2018, with a sampling interval of 5 minutes, and contains about 5.546 million traffic records. The data mainly includes fields such as time, street ID, traffic index, total travel length of the sample, average travel speed, and total travel time of the sample.

This paper uses the average travel speed (Speed) as the prediction target. Considering that the traffic operation status is affected by multiple factors, the traffic index (EXPONENT), total travel length of the sample (GOLEN), and total travel time of the sample (GOTIME) are used as auxiliary input variables to construct a multivariate time series prediction task together with historical speed information.

For time step t, the model input is represented as:

$$x_t = [v_t, l_t, \tau_t]T \tag{1}$$

Where v_t, e_t, l_t, and τ_t represent SPEED, EXPONENT, GOLEN, and GOTIME at time t, respectively.

$$\mathrm{X_t} = [\mathrm{x_{t-L+1}}, \mathrm{x_{t-L+2}}, \dots, \mathrm{x_t}] \in \mathrm{R^{L\times 4}} \tag{2}$$

The model predicts the average travel speed over multiple future time steps based on historical traffic conditions:

$$\mathrm{y_t} = [\mathrm{v_{t+1}}, \mathrm{v_{t+2}}, \dots, \mathrm{v_{t+H}}]\mathrm{T} \in \mathrm{R^H} \tag{3}$$

Where L = 576, H ∈ {6, 12, 24}, these correspond to the input over the past 48 hours and the predictions for the next 30, 60, and 120 minutes, respectively. Since urban traffic exhibits significant diurnal cyclical variations, this paper selects a 48-hour historical window, enabling the model to utilize both the traffic conditions of the previous day and recent trends.

Due to the large day-to-day cycle in urban traffic, in this study, a 48-hour historical period is used, so that the model can consider both the traffic conditions of the previous day and the current pattern. Based on the data sampling interval of 5 minutes, this study constructs three forecasting tasks for the next 30, 60, and 120 minutes, corresponding to the prediction step sizes of 6, 12 and 24 time slices, respectively, to investigate the variations of model prediction effectiveness over different forecasting scales.

The geographical relationships between various roads are not taken into consideration in this paper, which primarily concentrates on the temporal fluctuations of traffic factors. As a result, the topic falls under the category of multivariate time series forecasting.

2.2. Data preprocessing

First, TIME1 and PERIOD are combined into a DATETIME index with a 5-minute interval, and the average of duplicate records under the same BLOCKID and DATETIME is calculated. Then, the 75 road segments are aligned to a unified 5-minute time grid.

For short-term missing values, only historical observations are used for forward imputation, with a maximum imputation length of 12 time points; remaining missing values are imputed using the median of the corresponding road segment's training interval. Sliding window samples are constructed for each road segment, but they share the same set of model parameters, forming a global prediction model. The model input uses window-level instance normalization; the mean and standard deviation are calculated only from historical input windows, avoiding the use of future

DOI: 10.54254/2755-2721/2026.BA36282

target information. The training window is sampled every 12 time points, while the validation and test sets are sampled every 6 time points.

2.3. Baseline models

To investigate the forecasting capabilities of different deep learning architectures, this study selects LSTM and PatchTST as representative baseline models. LSTM represents recurrent neural network-based approaches, while PatchTST represents Transformer-based methods for long sequence forecasting. Both models are trained using identical input variables, forecasting targets, and experimental configurations to ensure a fair comparison.

2.3.1. LSTM

LSTM is adopted as the recurrent learning baseline in this study. Different from conventional recurrent neural networks, LSTM introduces a memory mechanism that enables the network to retain useful historical information during sequential processing. By updating hidden representations over time, LSTM can capture temporal variations within traffic speed sequences and generate future predictions based on previous observations.

In the traffic forecasting task, historical traffic states are provided as sequential inputs to the LSTM network. The hidden states generated during the encoding process are used to represent temporal characteristics, which are subsequently mapped to future average travel speed values.

2.3.2. PatchTST

PatchTST is selected as the Transformer-based forecasting model in this study. Instead of treating each timestamp as an independent input element, PatchTST transforms continuous observations into patch-level representations before applying Transformer encoders. This design reduces the length of the input sequence while preserving important temporal structures.

Through the attention mechanism, PatchTST is able to identify relationships among different temporal regions and capture both short-term fluctuations and long-range dependencies. Therefore, it provides an effective solution for modeling traffic sequences with extended historical inputs.

In this study, PatchTST is evaluated under the same experimental conditions as LSTM to analyze whether Transformer-based architectures can provide advantages in urban traffic speed forecasting tasks.

2.4. Model training strategy and assessment measures

The Mean Squared Error (MSE) loss function, which measures the discrepancy between the expected and actual vehicle speeds, is the optimization goal during model training. AdamW is the optimizer, with a batch size of 128, a maximum training epoch of 30, and an initial learning rate of 0.0005. The optimal model parameters for training. This study used mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and coefficient of determination (R2) as assessment indicators to thoroughly assess the prediction performance of the model. The error level between the prediction results and actual values is represented by MAE, RMSE, and MAPE; the lower the error value, the better the model's predictive ability. R^2 measures the model's ability to fit the trend of traffic speed changes; a value closer to 1 indicates stronger explanatory power.

DOI: 10.54254/2755-2721/2026.BA36282

3. Results and discussion

3.1. Experimental setup

This paper presents a comparative analysis of LSTM and PatchTST models for short-term traffic speed forecasting based on real-world traffic data collected from streets in Shenzhen, China.

For fair comparison, LSTM and PatchTST were trained and evaluated under the same experimental settings, including data partitioning, input variables, and forecasting targets. The dataset was divided chronologically into training, validation, and test sets with proportions of 70%, 15%, and 15%, respectively. Based on actual traffic forecasting needs, this paper sets three forecasting scales: 30 minutes, 60 minutes, and 120 minutes, corresponding to forecasting step sizes of 6, 12, and 24 time slices, respectively. The model input is a sequence of historical traffic states, and the output is the average travel speed over multiple future time steps. The MSE loss function was used to train the models. The AdamW optimizer was used with a batch size of 128, a maximum of 30 training epochs, and an initial learning rate of 0.0005. During training, early stopping was used, and the model parameters with the lowest validation loss were kept for the test set's final assessment.

In order to reduce the impact of random initialization on the experimental results, this paper uses three random seeds (42, 2024 and 3407) to repeat the experiment and reports the final test results.

The model structure parameters of the LSTM and PatchTST models are shown in Table 1.

Table 1. Model architecture parameters of LSTM and PatchTST

Model	Parameters	Settings
LSTM	Hidden Size	128
LSTM	Number of Layers	2
LSTM	Dropout	0.2
PatchTST	Sequence Length	576
PatchTST	Patch Length	24
PatchTST	Stride	12
PatchTST	Embedding Dimension	128
PatchTST	Attention Heads	4
PatchTST	Encoder Layers	3
PatchTST	Feed-forward Dimension	256
PatchTST	Dropout	0.2

The training parameters for the LSTM and PatchTST models are shown in Table 2.

Table 2. Unified training parameters of LSTM and PatchTST

Parameters	Settings
Input Features	4
Forecast Steps	6, 12, 24

DOI: 10.54254/2755-2721/2026.BA36282

Table 2. (continued)

Loss	MSE
Optimizer	AdamW
Learning Rate	0.0005
Weight Decay	0.0001
Batch Size	128
Maximum Epochs	30
Early Stopping Patience	5
LR Scheduler Patience	2
LR Decay Factor	0.5
Gradient Clipping	1.0

The experimental environment is shown in Table 3.

Table 3. Hardware and software environment settings

Item	Configuration / Environment
Platform	AutoDL
GPU	NVIDIA GeForce RTX 3080 Ti
GPU Memory	12 GB
CPU	12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz
Operating System	Ubuntu 22.04 LTS
Python Version	3.10
PyTorch Version	2.1.2
CUDA Version	11.8
Mixed Precision	Automatic Mixed Precision (AMP)

3.2. Comparison of prediction performance of different models

To evaluate the forecasting performance of LSTM and PatchTST under different forecasting horizons, experiments were conducted with prediction intervals of 30, 60, and 120 minutes. The results are presented in Table 4.

Table 4. Prediction performance comparison under different forecasting horizons

Forecast Horizon	Model	MAE	RMSE	MAPE(%)	R^2
30mins	LSTM	2.2223±0.0117	3.3826±0.0145	6.2777±0.0261	0.8764±0.0011
	PatchTST	2.0710±0.0043	3.2115±0.0040	5.8319±0.0179	0.8886±0.0003
60mins	LSTM	2.3834±0.0018	3.5508±0.0007	6.8243±0.0397	0.8637±0.0001
	PatchTST	2.1960±0.0071	3.3389±0.0040	6.2342±0.0181	0.8795±0.0003

DOI: 10.54254/2755-2721/2026.BA36282

Table 4. (continued)

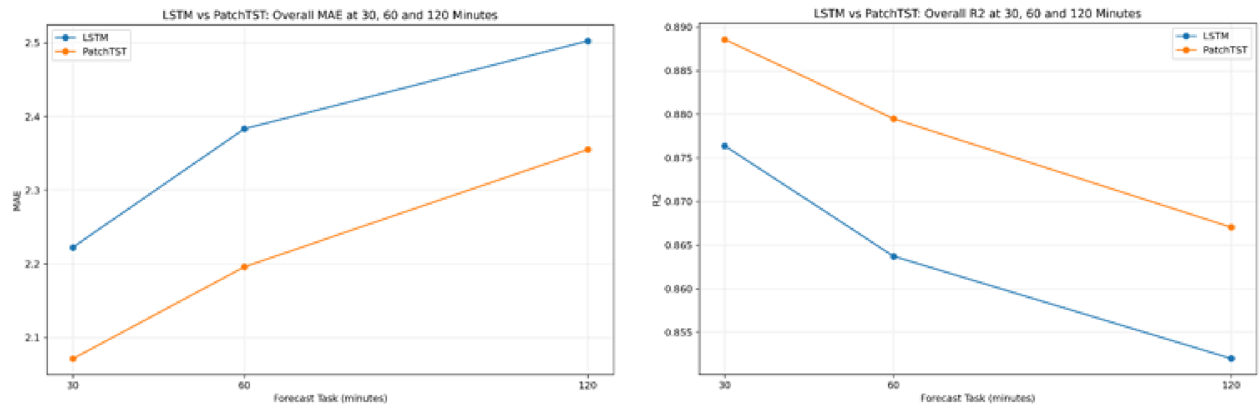
120mins	LSTM	2.5026±0.0280	3.6979±0.0331	7.2045±0.0687	0.8520±0.0026
	PatchTST	2.3552±0.0117	3.5049±0.0117	6.8043±0.0439	0.8670±0.0009

As shown in Table 4, both LSTM and PatchTST can effectively predict traffic speed change trends and have achieved good prediction results at different prediction scales. In the 30-minute prediction task, PatchTST achieved superior prediction performance, with MAE and RMSE of 2.0710 and 3.2115, respectively, lower than the LSTM model's 2.2223 and 3.3826. Simultaneously, PatchTST's R^2 reached 0.8886, higher than LSTM's 0.8764, indicating that PatchTST can more accurately fit the traffic speed variation pattern. As the prediction time range increases, the prediction errors of both models show an increasing trend. In the 120-minute prediction task, the MAE of LSTM and PatchTST reached 2.5026 and 2.3552, respectively, indicating that as the prediction distance increases, the uncertainty of traffic conditions increases, and the difficulty of model prediction further increases.

PatchTST outperformed LSTM in all three prediction tasks, with the largest relative improvement observed in the 60-minute task. As the prediction time domain increased from 30 minutes to 120 minutes, the errors and R^2 of both models increased, but PatchTST consistently maintained lower errors and higher fitting ability. This result indicates that PatchTST's advantage is stable across different prediction time domains, but its advantage does not monotonically increase with increasing prediction time domain.

3.3. Analysis of different prediction time scales

To further investigate the effect of forecasting horizons on model performance, the variations of evaluation metrics under different prediction intervals are analyzed.



(a) MAE (b) R^2

Figure 1. Comparison of performance metrics between LSTM and PatchTST under different forecasting horizons (photo/picture credit: original)

Figure 1 shows that as the prediction range increases from 30 minutes to 120 minutes, the prediction errors of both the LSTM and PatchTST models gradually increase. This is mainly because as the prediction distance expands, the future traffic conditions are more susceptible to random fluctuations and changes in traffic demand, leading to a decrease in the ability of historical information to constrain future conditions.

DOI: 10.54254/2755-2721/2026.BA36282

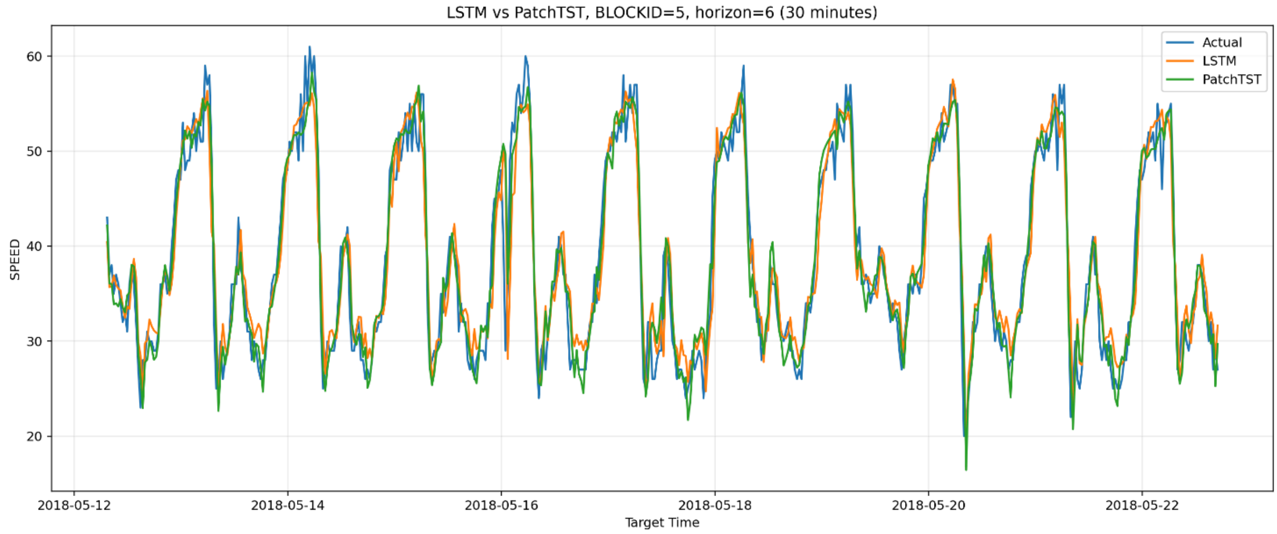


Figure 2. Comparison of predicted and ground truth traffic speeds between LSTM and PatchTST for a 30-minute forecasting horizon (photo/picture credit: original)

As shown in Figure 2, in short-term prediction tasks, traffic conditions have strong temporal continuity, and recent historical data can provide effective prediction information. Therefore, both models show good prediction capabilities.

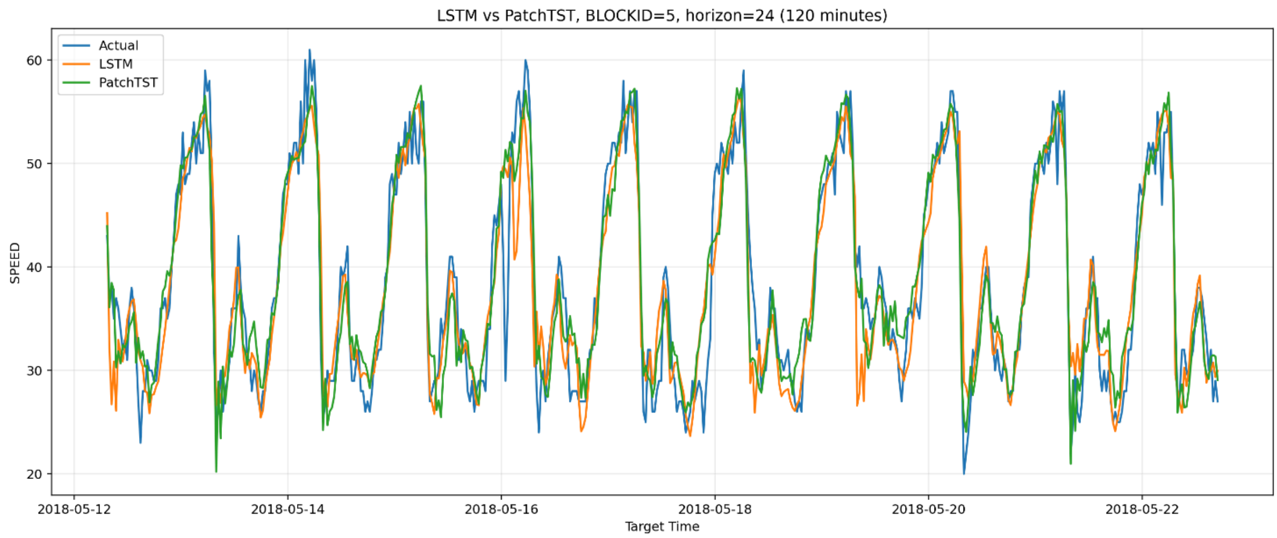


Figure 3. Comparison of predicted and ground truth traffic speeds between LSTM and PatchTST for a 120-minute forecasting horizon (photo/picture credit: original)

As illustrated in Figure 3, PatchTST achieves smaller prediction deviations than LSTM, particularly in long-term forecasting scenarios. When traffic speed changes rapidly, the prediction results of LSTM tend to show a certain delay compared with the actual values. In contrast, PatchTST responds more quickly to these variations and provides predictions that are closer to the observed trends.

The improved forecasting performance of PatchTST may be related to its Transformer-based structure. By dividing historical sequences into patches, PatchTST reduces the length of input tokens

DOI: 10.54254/2755-2721/2026.BA36282

and enables the Transformer encoder to capture temporal dependencies more efficiently through self-attention mechanisms.

Overall, LSTM and PatchTST both demonstrate the capability of modeling urban traffic speed variations. Compared with LSTM, PatchTST achieves more stable forecasting performance as the prediction horizon extends, indicating that its architecture is more suitable for capturing long-range temporal patterns in traffic data.

4. Conclusion

This study investigates the forecasting performance of LSTM and PatchTST for short-term urban traffic speed prediction using real-world street-level traffic data collected from Shenzhen, China. A multivariate forecasting framework is established by selecting average travel speed as the prediction target and incorporating traffic index, total travel length, and total travel time as additional input variables. Under identical experimental settings, both models are evaluated on three forecasting horizons, including 30, 60, and 120 minutes.

The experimental comparison shows that both LSTM and PatchTST can effectively model temporal variations in urban traffic speed sequences. However, PatchTST achieves consistently better performance across all forecasting horizons, with lower prediction errors and higher R^2 values. As the forecasting horizon increases, the prediction difficulty of both models becomes higher due to the growing uncertainty of future traffic conditions. Nevertheless, PatchTST maintains relatively stable performance, indicating stronger capability in capturing long-range temporal information.

The performance improvement of PatchTST is likely associated with its patch-based sequence representation and attention-based feature extraction mechanism. By reorganizing historical traffic observations into local temporal segments, PatchTST can reduce sequence complexity and improve the learning of both local variation patterns and global temporal relationships.

Several limitations remain in this study. The current framework mainly focuses on temporal characteristics of traffic sequences and does not explicitly model spatial interactions among different road segments. In addition, only traffic-related variables are considered, while external factors such as weather conditions, holidays, and unexpected events are not included.

Future research will explore spatiotemporal forecasting frameworks by integrating spatial modeling techniques, such as graph neural networks, to better represent dependencies among road segments. Moreover, incorporating additional external information and developing more efficient Transformer architectures may further improve forecasting accuracy and computational efficiency in complex urban traffic environments.

Author contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] 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.
[2] Lv, Y., Duan, Y., Kang, W., et al. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems, 16*(2), 865–873.
[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

DOI: 10.54254/2755-2721/2026.BA36282

[4] Ma, X., Tao, Z., Wang, Y., et al. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies, 54*, 187–197.

[5] Wu, P., Huang, Z., Pian, Y., et al. (2020). A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting. *Journal of Advanced Transportation, 2020*, Article 8863724.

[6] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).

[7] Nie, Y., Nguyen, N. H., Sinthong, P., et al. (2023). A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[8] Wu, H., Hu, T., Liu, Y., et al. (2023). TimesNet: Temporal 2D-variation modeling for general time series analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[9] Liu, Y., Hu, T., Zhang, H., et al. (2024). iTransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[10] Shenzhen Municipal Bureau of Transport. (2021, April 21). Street real-time data [Data set]. *Shenzhen Open Data Portal*. https://opendata.sz.gov.cn/data/dataSet/toDataDetails/29200_00403589