Hybrid Deep Learning Model for Traffic Flow and Speed Forecasting at Urban Junction

Li Ma[1,2,a*], Yongshun Wang[1,b], Xiaoshi Lv[3,c], Yongheng He[2,d], Guixin Wang[2,e]

[1] School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China;

[2] School of Electronic and Electrical Engineering, Lanzhou Petrochemical Vocational and Technical University, Lanzhou 730060, China;

[3] CCCC (Xi'an) Railway Design & Research Institute Co., Ltd, Xi'an 710065, China;

a mali@lzpcc.edu.cn, b wangysh@mail.lzjtu.cn, c 17793192475@163.com

d 2601604954@qq.com, e 3020882298@qq.com

Abstract. The hybrid deep learning model H-CNN-LSTM (Hybrid Convolution Neural Network and Long Short-Term Memory) is introduced in this paper. The model combines CNN to extract traffic flow and speed features, LSTM networks to model temporal dependencies, and employs a combined matrix and a rolling forward input method during data input, enhancing the coarse-grained extraction of traffic data and further refining the utilization of adjacent data. Our research focuses on the complex dynamic traffic conditions of Fuli East Road in Xigu District, Lanzhou City of China. Using detailed traffic video data collected during one month of evening rush hours, we predict future traffic using the H-CNN-LSTM model. By comparing it with other prediction methods, we validate the effectiveness of our approach in predicting traffic flow and speed in real-world traffic scenarios.

Keywords: Intelligent Transportation; smart city; traffic Prediction; deep Learning.

1. Introduction

Traffic congestion and variability are not mere inconveniences; they are critical issues that permeate the fabric of urban life, exerting substantial pressure on the economic vitality, environmental health, and overall well-being of city dwellers. In the bustling arteries of modern cities, the persistent pulse of vehicular movement forms a complex web of interactions. This web is a product of countless variables: from the rhythmic patterns of daily commutes to the unexpected incidents that disrupt the flow, from the seasonal changes that reshape travel behaviours to the infrastructural limitations that constrain movement. The intricate dance of urban traffic is a reflection of the city's living, breathing ecosystem. In such an environment, effective traffic management transcends the act of mere regulation; it becomes a dance of its own, requiring a predictive grasp of the rhythm and flow of traffic. The ability to foresee how traffic will move, constrict, and expand is essential for preempting bottlenecks, optimizing traffic light sequences, planning road works, and implementing dynamic routing. However, traffic prediction is a problematic challenge, teeming with volatility and complexity.

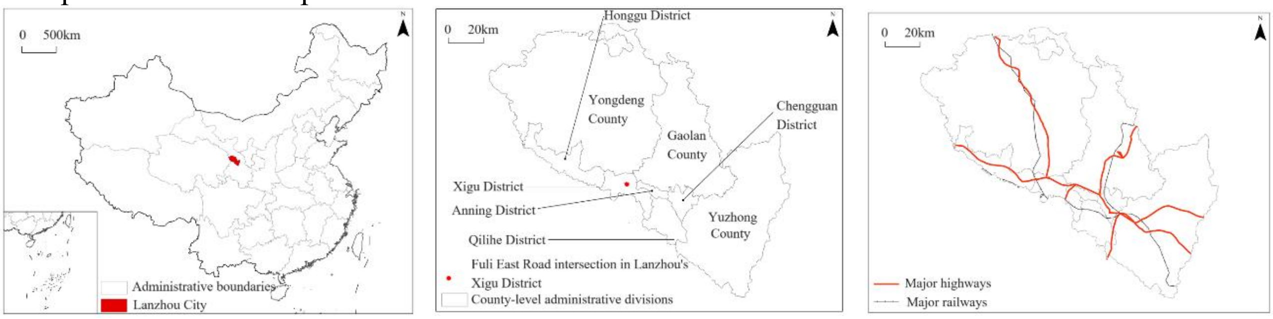
The landscape of traffic modelling and forecasting is both diverse and complex, with researchers exploring a wide array of methodologies to tackle the inherent challenges of predicting traffic flows and congestion patterns. Convolutional Neural Networks (CNN) apply grid-based traffic data to capture spatial dependencies, such as STResNet[1], RSTS[2]. The transfer learning method[3] and the wavelet neural network[4], which are used in traffic flow prediction. Recurrent Neural Networks (RNN) are employed to learn temporal dynamics, such as LSTM[5], DSAN[6], AGCRN[7], and the asynchronous residual network[8]. Later, Graph Neural Networks (GNN) became more suitable for modeling the underlying graph structure of traffic data. Therefore, GNN-based methods have become widely popular in traffic prediction, such as GWNE[9] and STGNCDE[10]. With the emergence of Transformer-based models, such as PDFormer[11], STAEformer[12], STANN[13], and the later popular spatiotemporal dependencies[14], these models have also gained attention in the field.

The CNN-LSTM neural network[15] combines the feature recognition capabilities of CNN with the temporal processing capabilities of LSTM networks, achieving good results in traffic flow prediction. Based on this, we have professionally designed the CNN layer to interpret time series data as multi-dimensional feature matrices, thereby identifying and extracting complex relationships between various traffic indicators. By using a sliding input approach, we enhance the extraction of fine-grained traffic features. This is then seamlessly integrated with the LSTM layer. The resulting hybrid architecture can be used to learn and predict future traffic conditions at specified locations with extremely high accuracy.

2. Data

2.1 Research area

Our study area is Lanzhou, a central city in Northwest China, known for its significant industrial base and role as a transportation hub on the Silk Road Economic Belt. The Xigu District, the focus of our study, presents a unique set of traffic dynamics influenced by its position as a petrochemical base and its intersection with major transportation routes like the Lanzhou-Xinjiang Railway. The Fuli East Road intersection, in particular, is characterized by high vehicle flow and intricate traffic conditions, representing a microcosm of the broader urban traffic scenarios found in rapidly developing cities. Figure 1(a) is a map of China, Figure 1(b) is a map of Lanzhou, and Figure 1(c) is a map of the main transportation routes in Lanzhou.



(a) China (b) Lanzhou (C) Main transportation routes in Lanzhou

Fig. 1 Study area map

2.2 Data collection

We undertook a meticulous data collection process at the Fuli East Road intersection, located in the industrially bustling Xigu District of Lanzhou. This data collection spanned a one-month period, commencing on June 4 and concluding on July 3, 2022. Our focus was specifically on the evening rush hour, a critical timeframe for traffic flow analysis, which we defined as the interval from 17:30 to 19:00 each day. Throughout this period, we systematically captured video footage of the intersection, resulting in an extensive archive of 30 traffic videos, all formatted in .MP4. To ensure consistency in temporal analysis, speed and flow data were aggregated into 10-second intervals, resulting in a dataset that provides a detailed representation of the traffic conditions during rush hour.

2.3 Data processing

As shown in Fig. 2, in the data processing stage of the research, we adopted a systematic method to convert the original traffic video data into a structured format suitable for analysis. The processing steps are as follows:

Vehicle Detection and Tracking: Initially, our imaging system searched for and identified vehicles within the videos. Once detected, the vehicles were continuously tracked throughout the video sequence to monitor their trajectory and behavior.

Speed Tracking and Calculation: The system tracked each vehicle's speed by calculating the change in its position over time. If a vehicle's speed at time 'i' was higher than at time 'i-1', tracking continued; otherwise, the speed was recorded.

Data Refinement: To derive the final fundamental traffic data, we implemented a two-step refinement process. We calculated the average speed of three adjacent data points to smooth the data and then selected the maximum of these averages to represent the vehicle's speed. This approach helped to mitigate short-term fluctuations and anomalies in speed.

Outlier Detection and Removal: Outliers, which could arise from sudden vehicle movements or tracking errors, were identified and eliminated to ensure the integrity of our dataset.

Final Data Recording: After processing, the cleaned and validated speed data were recorded, yielding a comprehensive dataset that transformed the raw video into tens of thousands to over 200,000 actionable data records.

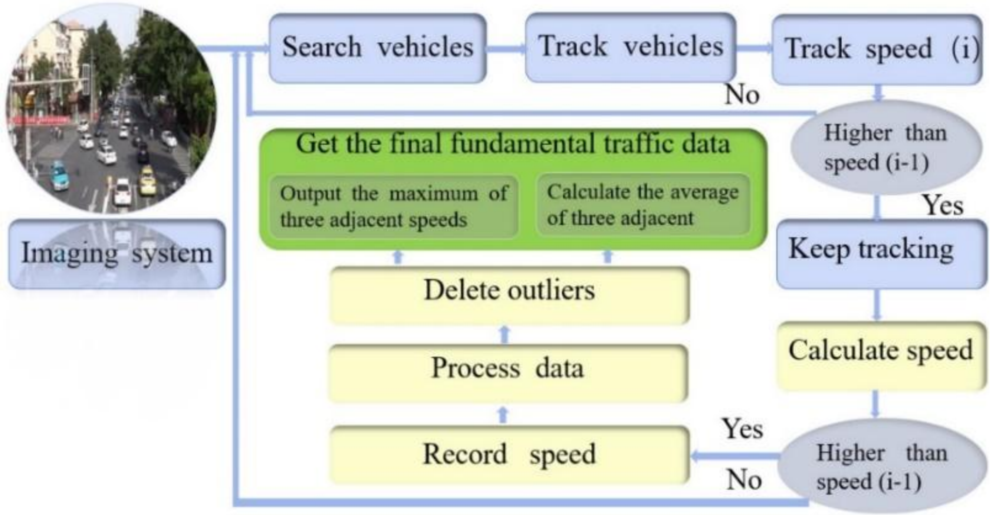


Fig. 2 Data transformation procedure

The raw speed and flow data, derived from the vehicle tracking procedure, initially presented a set of discrete measurements for each vehicle at varying time steps. This granularity, while precise, resulted in an irregular temporal distribution with potential overlaps of multiple data points within a single time step. To address this and standardize the dataset, we implemented a temporal aggregation technique. Speed and flow data were consolidated into uniform intervals of 10 seconds. Within each interval, we calculated the average speed of all vehicles tracked during that period. This method effectively synthesized the variable data points into a single representative value per time step, thereby smoothing out the fluctuations.Through this process of temporal averaging, we transformed the dataset into a structured sequence of 540 data points, corresponding to the 90-minute rush hour period. Each data point represents the aggregate traffic speed and flow within its respective 10-second window, yielding a dataset amenable to robust time-series analysis and predictive modeling.

3. H-CNN-LSTM Model

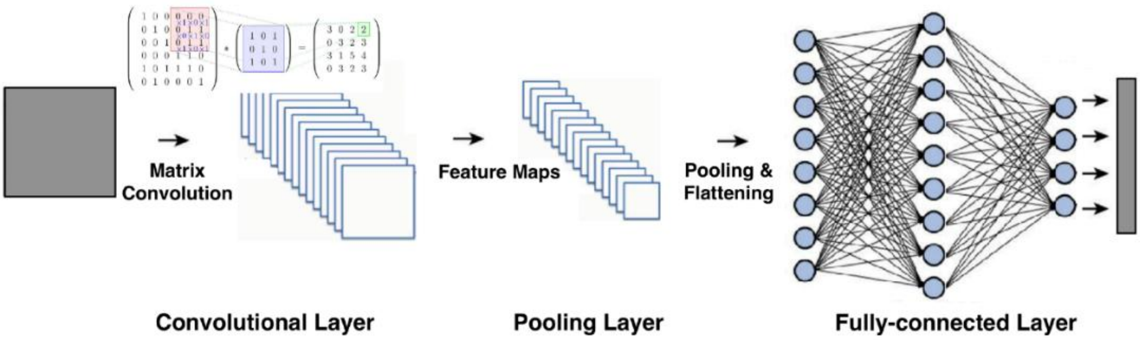


Fig. 3 Schematic diagram of H-CNN architecture

As shown in Figure 3, the input historical traffic flow data is constructed into a matrix Y that contains traffic flow feature vectors, defined as:

$$Y=\begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & y_1 \\ x_1^2 & x_2^2 & x_3^2 & y_2 \\ x_1^3 & x_2^3 & x_3^3 & y_3 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^i & x_2^i & x_3^i & y_i \end{bmatrix} (i=1,2,3\cdots n) \tag{1}$$

In the formula: x_1^irepresents the real-time speed at time i,x_2^irepresents the maximum speed at time i, x_3^irepresents the average speed at time i, and y_i represents the traffic flow at time i. All of these data are historical data.

Input the feature matrix Y into a multi-layer CNN to extract traffic flow data features, and obtain the traffic flow sequence feature vector Z:

$$Z=\begin{bmatrix} Y_{i1} & Y_{i2} & Y_{i3} & \cdots & Z_{i1} \\ Y_{m1} & Y_{i4} & Y_{i5} & \cdots & Z_{m1} \\ Y_{m2} & Y_{i6} & Y_{i7} & \cdots & Z_{m2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ Y_{mn} & Y_{i(n+1)} & Y_{i(n+2)} & \cdots & Z_{mn} \end{bmatrix} (i=1,2,3\cdots n;\ m=1,2,3\cdots n) \tag{2}$$

In the formula: i represents the moment of data features, m represents the number of iterations for data prediction, and n represents the moment of data features after iteration.

The convolution operation in layer l for a given filter k is defined as:

$$Z^{[l](k)} = W^{[l](k)} * A^{[l-1]} + b^{[l](k)} \tag{3}$$

Where $*$ denotes the convolution operation, $W^{[l](k)}$ is the weight matrix of the k-th filter in the l-th layer, $A^{[l-1]}$ is the activation from the previous layer, and $b^{[l](k)}$ is the bias term for the k-th filter. Next, the Rectified Linear Unit (ReLU) activation function is applied element-wise:

$$A^{[l](k)} = \max\left(0, Z^{[l](k)}\right) \tag{4}$$

Where $A^{[l](k)}$ is the activation of the k-th filter at layer l after applying ReLU to the convolved feature $Z^{[l](k)}$. The pooling operation is applied to reduce the spatial size:

$$P^{[l](k)} = \mathrm{MaxPool}\left(A^{[l](k)}\right) \tag{5}$$

Where $P^{[l](k)}$ is the result of applying the max pooling operation to activation $A^{[l](k)}$. The fully connected layer flattens the pooled features and then performs a linear transformation:

$$F^{[l]} = W^{[l]} \cdot \mathrm{Flatten}\left(P^{[l-1]}\right) + b^{[l]} \tag{6}$$

Where $F^{[l]}$ is the output of the fully connected layer l, $W^{[l]}$ and $b^{[l]}$ are the weights and biases, and $P^{[l-1]}$ are the pooled features from the previous layer. The softmax function is often used in the output layer for multi-class classification tasks:

$$\mathrm{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \tag{7}$$

Where z is the input vector to the softmax function, z_i is the i-th element of vector z, and K is the number of classes.

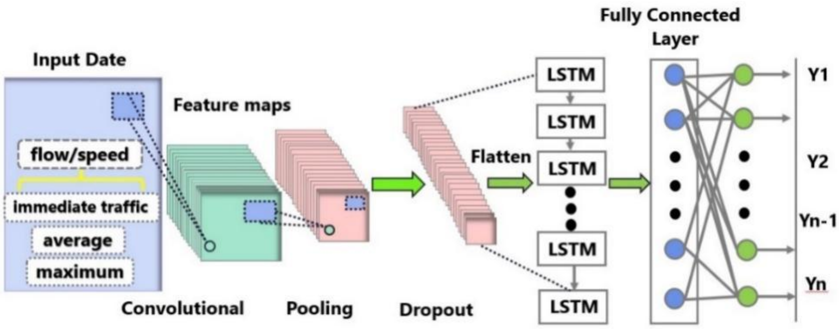


Fig. 4 H-CNN-LSTM model architecture diagram

As shown in Figure 4, The underlying mechanism of an LSTM network includes three principal layers: the input, hidden, and output layers. Its structure consists of a cell that conveys information through the sequence and three regulatory gates: forget, input, and output gates.The forget gate processes inputs from two sources: the preceding layer's output, denoted as h_{t-1}, and the current input X_t. Following this, the input gate determines the significance of new information and updates the cell state accordingly. This gate utilizes sigmoid and tanh activations, with the sigmoid activation determining the information's relevance and the tanh function modulating the network bias. Lastly, the output gate is responsible for producing the next hidden state h_t, by filtering the information through a sigmoid function and then modulating the updated cell information from the cell state with a product of the tanh output and the sigmoid function's output. The processes within the LSTM unit are mathematically articulated as follows.

Forget gate:

$$f_t = \sigma(W_f \times X_t + V_f \times h_{t-1} + b_f) \tag{8}$$

Input gate:

$$i_t = \sigma(W_i \times X_t + V_i \times h_{t-1} + b_i) \tag{9}$$

$$\widetilde{C}_t = \tanh(W_{\tilde{C}} \times X_t + V_{\tilde{C}} \times h_t + b_{\tilde{C}}) \tag{10}$$

Cell state update:

$$C_t = f_t \times C_{t-1} + i + t \times \widetilde{C}_t \tag{11}$$

$$o_t = \sigma(W_o \times X_t + V_o \times h_{t-1} + b_o) \tag{12}$$

$$h_t = o_t \times \tanh(C_t) \tag{13}$$

Where X_t denotes the input data at time step t and C_t indicates the hidden cell state at t. Variables f, i, C and o represent forget gate, input gate, cell state vectors, and output gate, respectively. Operator $\times$ represents the scalar product of two vectors, $\sigma()$ denotes the standard logistic sigmoid function, which scales the inputs into the range of $(0,1)$, and $\tanh()$ is the hyperbolic tangent function, which transforms the input and output into the range of $[-1,1]$. Parameters $W_j(j = f, i, \tilde{C}, o)$ and $V_j(j = f, i, \tilde{C}, o)$ denote the weight matrices in each layer and $b_j(j = f, i, \tilde{C}, o)$ denotes the bias of each gate.

Our model ingests a structured sequence of data where each entry is a time step containing three distinct features: the immediate traffic flow/speed, the historical average flow/speed within a designated time window, and corresponding timestamps. This multi-feature approach captures various aspects of traffic dynamics, providing a nuanced view of the traffic state. By predicting multiple steps ahead, the final hidden state h_n from the LSTM is then passed through a fully connected layer to produce the output Y:

$$Y = W_{fc} \cdot h_n + b_{fc} \tag{14}$$

Where W_{fc} and b_{fc} are the weights and biases of the fully connected layer. The model concludes with an output layer that produces a future sequence of traffic flow/speed predictions, corresponding to the next couples of time steps. This output provides a tangible forecast that can inform traffic control measures and planning.

4. Experimental Study

4.1 Model Training and Optimisation

To train H-CNN-LSTM model, we define the performance of our model using the Mean Squared Error (MSE) as the loss function, which is formulated as follows for a set of m training examples:

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{Y}^{(i)} - Y^{(i)}\right)^2 \tag{15}$$

Where $\hat{Y}^{(i)}$ represents the predicted output of the model, and $Y^{(i)}$ is the true output for the i-th example. The Adam optimizer is selected for its effectiveness in handling sparse gradients and its adaptive learning rate capabilities. Adam combines the advantages of two other extensions of stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Adam calculates an exponentially weighted average of past gradients and past squared gradients, and the parameters β_1 and β_2 control these moving averages. The parameter update rules for the Adam optimizer at time step t are given by:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}\hat{m}_t \tag{16}$$

Where: θ represents the parameters of the model, η is the learning rate, $\hat{m}_t$ is the bias-corrected first moment estimate, $\hat{v}_t$ is the bias-corrected second raw moment estimate, and ϵ is a small scalar used to prevent division by zero.

4.2 Performance Measurement Metric

To evaluate the performance of our predictive model, we employed three widely recognized metrics: Mean Squared Error (MSE) (as formed in Eq. 15), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). These metrics are formally defined as follows:

Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{17}$$

Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \tag{18}$$

Where n is the number of observations, Y_i is the actual value, and $\hat{Y}_i$ is the predicted value.

4.3 Comparison Models

To validate the performance of our hybrid deep learning model comprehensively, we compared it with three prominent models: Feedforward Neural Networks (FNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. Each of these models was adapted to use the same input data format and aimed to predict the same output format as our proposed model, ensuring a consistent and fair comparison across all models. The FNN model, with its straightforward architecture, tests the baseline performance in traffic flow and speed prediction without the complexities of spatial or temporal feature extraction. The CNN model, renowned for its efficacy in extracting hierarchical spatial features, was evaluated for its ability to understand the spatial patterns within the traffic data. Meanwhile, the LSTM model, celebrated for its ability to capture long-term dependencies, was included to assess the impact of temporal dynamics on prediction accuracy. Through this comparative analysis, we aim to demonstrate the superior ability of our hybrid model to integrate and leverage both spatial and temporal data features effectively, thereby providing a more accurate and robust prediction of traffic conditions at urban intersections.

4.4 Performance Results

In this section, we delve into a comparative analysis of the performance of various deep learning models applied to traffic speed and flow prediction. The models were assessed using normalized traffic datasets, with values scaled between 0 and 1 to maintain uniformity and comparability. The evaluation metrics used—Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE)—provide a multi-faceted view of each model's accuracy, consistency, and reliability.

Table 1 Performance comparison for traffic speed prediction on weekday and weekend datasets

Model	Weekday Dataset			Weekend Dataset		
	MSE	RMSE	MAE	MSE	RMSE	MAE
FNN	0.20%	2.98%	1.92%	0.28%	4.86%	3.79%
CNN	0.13%	2.11%	1.14%	0.27%	4.66%	3.74%
LSTM	0.11%	1.19%	0.28%	0.22%	4.05%	2.99%
H-CNN-LSTM	0.11%	0.94%	0.15%	0.03%	1.79%	1.35%

Analyzing the performance on traffic speed prediction, the results as shown in Table 1 indicate that the hybrid H-CNN-LSTM model consistently outperforms other models in both the weekday and weekend datasets. On weekdays, the H-CNN-LSTM model achieves the lowest MSE (0.11%) and RMSE (0.94%), indicating a strong predictive accuracy. The low MAE (0.15%) during weekdays suggests that the model's predictions are very close to the actual speeds. During weekends, the model also achieves exceptional performance with the lowest errors across all metrics, reflecting its robustness to the variability of traffic patterns on weekends.

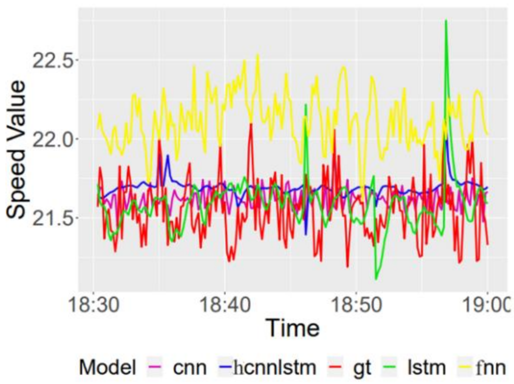
Table 2 Performance comparison for traffic flow prediction on weekday and weekend datasets

Model	Weekday Dataset			Weekend Dataset		
	MSE	RMSE	MAE	MSE	RMSE	MAE
FNN	3.27%	17.48%	14.74%	2.42%	15.35%	12.49%
CNN	0.49%	6.97%	5.52%	1.16%	10.62%	8.77%
LSTM	1.14%	10.66%	8.48%	1.54%	12.25%	9.63%
H-CNN-LSTM	0.39%	6.21%	4.68%	0.50%	7.02%	5.30%

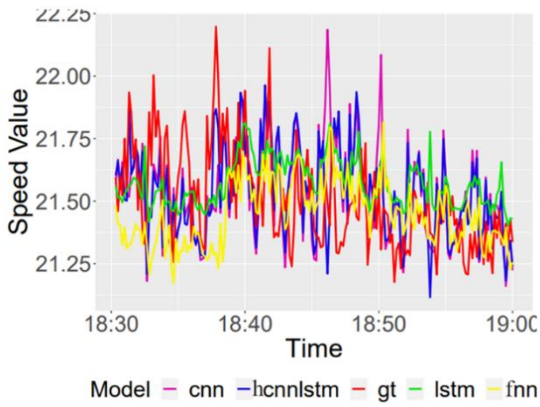
Table 2 presents the results for traffic flow prediction, where once again, the H-CNN-LSTM model demonstrates superior performance across both datasets. It achieves remarkably low MSE (0.39%) for weekday and 0.50% for weekend, suggesting a high level of precision. The RMSE is also the lowest for the H-CNN-LSTM model, especially in weekend dataset (7.02%), which indicates reliable predictive capability even when dealing with more fluctuating traffic volumes.

4.5 Case

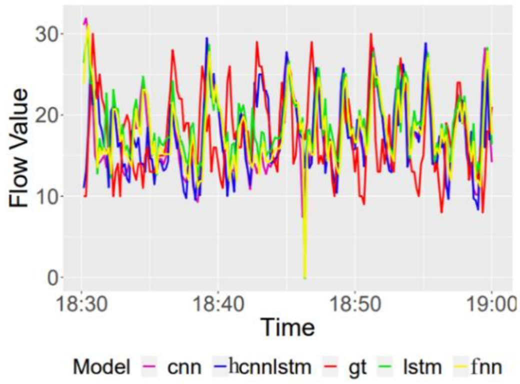
In this case study, We focus on a specific urban intersection in Xigu District, Lanzhou, and reflect real-world conditions by distinguishing between weekday and weekend traffic patterns, which exhibit distinctly different characteristics.Ground truth (gt) data, representing the actual observed traffic conditions, serves as a benchmark for assessing the predictive accuracy of each model.



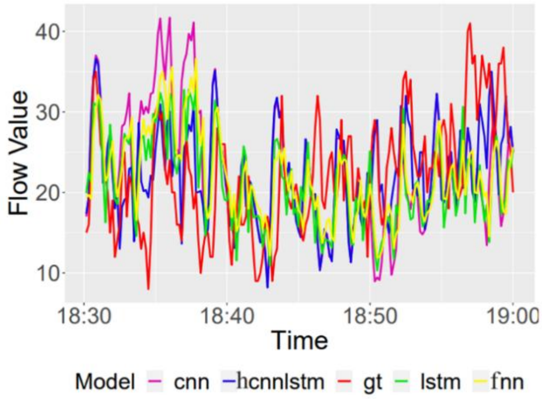
(a) Weekday speed prediction



(b) Weekend speed prediction



(c) Weekday flow prediction



(d) Weekend flow prediction

Fig.5 Case study on prediction results for one specific weekday and weekend respectively

For the weekday speed prediction, as shown in Figure 5 (a), the H-CNN-LSTM model closely tracks the ground truth with minimal deviation, indicating its robust predictive capability in capturing the traffic speed patterns during regular weekday traffic conditions. The LSTM model also shows a commendable performance, although with slightly higher variance from the ground truth compared to H-CNN-LSTM.

In the weekend speed prediction scenario, Figure 5 (b) reveals that all models face increased difficulty in tracking the ground truth, reflecting the inherent unpredictability and variation in weekend traffic patterns. Despite this, the H-CNN-LSTM model maintains the closest approximation to the ground truth, outperforming the standalone LSTM and FNN models, which exhibit greater fluctuations and thus, a higher degree of inaccuracy.

Moving on to the weekday flow prediction in Figure 5 (c), the H-CNN-LSTM and LSTM models demonstrate a tight convergence with the ground truth, illustrating their ability to understand and predict traffic flow during the structured patterns of weekday traffic. It is evident that temporal features play a significant role in predicting flow, as models equipped with LSTM cells tend to follow the ground truth more closely.

The weekend flow prediction shown in Figure 5 (d) displays a similar trend, with the H-CNN-LSTM model providing the most accurate predictions. The LSTM model still performs well, but the added spatial analysis from the CNN layers in the hybrid model appears to give it an edge in handling the complexities of weekend traffic flow.

5. Summary

In this paper, we have tackled the complex problem of urban traffic prediction by employing a hybrid deep learning approach, integrating the spatial feature extraction capabilities of CNNs with the temporal dynamism of LSTMs. Our focus on Lanzhou's Xigu District, particularly the Fuli East

Road intersection, has provided us with a rich dataset reflective of the multifaceted nature of urban traffic flow and speed. Through meticulous data collection and preprocessing, we have transformed raw traffic video data into a structured format conducive for deep learning algorithms. The proposed H-CNN-LSTM model has demonstrated a high degree of accuracy in predicting traffic conditions, outperforming standard FNN, CNN, and LSTM models in both weekday and weekend scenarios. The results affirm the model's adeptness at capturing the inherent variabilities of traffic patterns, thus showcasing its practical utility for real-time traffic management and planning.

For future work, we propose to explore the direct incorporation of video data into the model to potentially enhance the granularity and richness of the input data. By utilizing raw video footage, the model may be able to learn more intricate patterns and subtleties in traffic behaviour that are not captured in aggregated data. Additionally, we aim to extend our model to multi-step predictions that can forecast traffic conditions over a longer term. This extension would allow traffic management systems to plan more effectively by anticipating future traffic scenarios and adjusting measures proactively. As urban environments continue to grow in complexity, the development of advanced predictive models such as the one proposed in this study will become increasingly vital for maintaining the smooth operation of city infrastructures.

Acknowledgements

National Natural Science Foundation of China project(6136606), Innovation Fund Project for University Teachers of Gansu Provincial Education Department(2023B-294)

References

[1] Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]. Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1).

[2] Yao H, Tang X, Wei H, et al. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction[C].Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 5668-5675.

[3] Ren, Y; Chen, X; Wan S, et al. Passenger flow prediction in traffic system based on deep neural networks and transfer learning method[C].2019 4th International Conference on Intelligent Transportation Engineering (ICITE): 2019,115-120.

[4] Wang, d; Zhang, Q.; Chen, Y.; Ye, Z. An urban short-term traffic flow prediction model based on wavelet neuralnetwork with improved whale optimization algorithm[J]. Sustainable Cities and Society,2021,102858

[5] Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. Advances in neural information processing systems, 2015, 28.

[6] Lin H, Bai R, Jia W, et al. Preserving dynamic attention for long-term spatial-temporal prediction[C]. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 36-46.

[7] Bai L, Yao L, Li C, et al. Adaptive graph convolutional recurrent network for traffic forecasting[J]. Advances in neural information processing systems, 2020, 33: 17804-17815.

[8] GAO Hai-long,XU Yi-bo,HOU De-zao, et al. Short-term traffic flow prediction algorithm for road network based on deep asynchronous residual network[J]. Journal of Jilin University (Engineering and Technology Edition), 2023,53(12):3458-3464.

[9] Wu Z, Pan S, Long G, et al. Graph WaveNet for deep spatial-temporal graph modeling[C]. International Joint Conference on Artificial Intelligence 2019. Association for the Advancement of Artificial Intelligence (AAAI), 2019: 1907-1913.

[10] Choi J, Choi H, Hwang J, et al. Graph neural controlled differential equations for traffic forecasting[C]. Proceedings of the AAAI conference on artificial intelligence. 2022, 36(6): 6367-6374

[11] Jiang J, Han C, Zhao W X, et al. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction[C].Proceedings of the AAAI conference on artificial intelligence. 2023, 37(4): 4365-4373.

[12] Liu H, Dong Z, Jiang R, et al. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting[C]. Proceedings of the 32nd ACM international conference on information and knowledge management. 2023: 4125-4129.

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

[14] Zhang, Z; Li, M; Lin, X; Wang, Y; and He,F. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. Trans-portation research part C: emerging technologies, 2019, 105:297–322.

[15] Bogaerts, T.; Masegosa, A.D.; Angarita,Z. J. et al. Agraph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data[J].Transportation Research Part C:Emerging Technologies, 2020,112:62-77.