Time Series-Based Highway Traffic Prediction

Deep Learning Lesson Project Final Report

Abstract

This study aims to predict traffic volume on Minnesota's Interstate 94 using the Metro Interstate Traffic Volume dataset. Data preprocessing includes handling missing values, correcting outliers, normalizing features, encoding categorical variables, and analyzing trend, stationarity, and autocorrelation. We implemented and evaluated four deep learning models: Long Short-Term Memory (LSTM), Bidirectional LSTM with Attention (BiLSTM+Attention), Convolutional Neural Network-LSTM (CNN-LSTM), and Temporal Convolutional Network (TCN). Experimental results show that LSTM achieved the best performance on the test set with a loss of 0.4759, demonstrating strong generalization. BiLSTM+Attention attained the lowest training loss of 0.4815 but showed signs of overfitting. CNN-LSTM exhibited the most balanced performance, achieving the lowest test loss of 0.4645. Despite TCN's high training and validation losses, it performed well on the test set with a loss of 0.4525. The CNN-LSTM model stands out for effectively capturing temporal dependencies and local patterns, offering a robust solution for traffic volume prediction. This study highlights the potential of deep learning in time series forecasting and provides a foundation for future research, including model optimization and the integration of external features to enhance predictive accuracy.

1. Introduction

Traffic volume prediction is a critical component of intelligent transportation systems, playing a pivotal role in traffic management, infrastructure planning, and congestion mitigation. With the rapid growth of urbanization and the increasing complexity of transportation networks, accurate and reliable traffic forecasting has become more essential than ever. Traditional methods, such as statistical models and machine learning approaches, have been widely used for time series prediction. However, these methods often struggle to capture the intricate temporal dependencies and nonlinear patterns inherent in traffic data, particularly when influenced by external factors such as weather conditions, holidays, and special events.

In recent years, deep learning has revolutionized time series forecasting, with models like LSTM, GRU, Transformer, CNN, TCN, and GNNs demonstrating unique strengths in capturing temporal dependencies, local patterns, and spatial relationships. Despite these advancements, challenges remain in traffic volume prediction, particularly in handling the complex interplay of external factors. Existing studies often focus on single models or lack comprehensive comparisons, leaving gaps in understanding their relative performance and applicability. Moreover, hybrid architectures and attention mechanisms, which could enhance model robustness and accuracy, are underexplored in this domain.

Given the limitations of individual models in terms of functionality and focus areas, our research attempts to integrate several basic deep learning models to enhance prediction accuracy under multiple complex factors.This study addresses these gaps by systematically evaluating LSTM, BiLSTM+Attention, CNN-LSTM, and TCN for traffic volume prediction, aiming to identify the most effective approach for capturing both long-term dependencies and localized patterns. By doing so, this research not only advances methodological frameworks for traffic forecasting but also provides practical insights for real-world applications in intelligent transportation systems.

2. Related work

Traffic volume prediction, as a core issue in intelligent transportation systems, has garnered significant attention in recent years. Researchers have proposed various methods from different perspectives, which can be broadly categorized into traditional statistical methods, machine learning methods, and deep learning methods. Traditional statistical methods dominated early research, primarily including Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA) models. For example, Williams and Hoel (2003) used the ARIMA model to predict short-term traffic flow, achieving promising results [1]. However, these methods rely on linear assumptions, making it difficult to capture nonlinear features and complex temporal dependencies in traffic data. Additionally, the Prophet model proposed by Smith et al. (2012) can handle seasonality and holiday effects but performs poorly when dealing with high-dimensional features and external factors such as weather [2]. The common limitation of these methods lies in their strong assumptions about data distribution and reliance on feature engineering, which restricts their generalization ability in practical applications.

With the advancement of machine learning techniques, methods such as Support Vector Machines (SVM), Random Forests (RF), and Gradient Boosted Decision Trees (GBDT) have been introduced into the field of traffic volume prediction. For instance, Zhang and Xie (2008) employed SVM to predict traffic flow, using kernel functions to address nonlinear relationships, achieving better results than traditional statistical methods [3]. However, these methods require manual feature extraction and perform poorly when handling long-term time series data. In recent years, ensemble learning methods such as XGBoost and LightGBM have also been applied to traffic prediction. For example, Chen et al. (2019) used XGBoost combined with multi-source data (e.g., weather and holidays) to significantly improve prediction accuracy [4]. Nevertheless, machine learning methods still face challenges such as high computational complexity and poor interpretability when dealing with high-dimensional time series data.

The rise of deep learning has provided new solutions for traffic volume prediction. Long Short-Term Memory (LSTM) networks have been widely adopted due to their ability to capture long-term dependencies. For example, Yu et al. (2017) used LSTM to predict urban traffic flow, achieving significantly better results than traditional methods [5]. However, LSTM suffers from issues such as gradient vanishing and low computational efficiency when processing long sequences. To address these limitations, researchers have proposed various improvements, such as Gated Recurrent Units (GRU) [6] and Temporal Convolutional Networks (TCN). For instance, Bai et al. (2018) introduced TCN, which significantly enhances long-sequence modeling capabilities through dilated convolutions and multi-layer structures[7]. Additionally, the introduction of attention mechanisms has further improved the model's ability to focus on critical time steps. For example, the Transformer model proposed by Vaswani et al. (2017) has demonstrated excellent performance in long-sequence prediction tasks[8]. Despite these advancements, existing research often focuses on single models or specific scenarios, lacking systematic comparisons of multiple deep learning models, particularly in the context of traffic volume prediction.

Despite the achievements of the aforementioned methods in traffic volume prediction, several limitations remain. These include the predominance of single-model studies, insufficient utilization of external features (e.g., weather and holidays), overfitting issues in certain models (e.g., BiLSTM+Attention), and low computational efficiency in some approaches (e.g., Transformer). To address these gaps, this study proposes a deep learning-based framework for traffic volume prediction, systematically comparing the performance of four models: LSTM, BiLSTM+Attention, CNN-LSTM, and TCN. The key innovations of this research include multi-model comparison, hybrid architectures, attention mechanisms, and efficient modeling. Through these innovations, this study not only fills existing research gaps but also provides efficient and reliable solutions for traffic volume prediction, laying a theoretical foundation for practical applications in intelligent transportation systems.

3. Method

This section elaborates on the deep learning-based traffic volume prediction methods proposed in our study. Our research primarily compares four models: the traditional LSTM network, Bidirectional LSTM with an attention mechanism , a hybrid model combining CNN and LSTM CNN-LSTM, and TCN. Through systematic comparison, we aim to analyze the performance of these models in traffic volume prediction tasks and reveal their respective strengths and limitations.

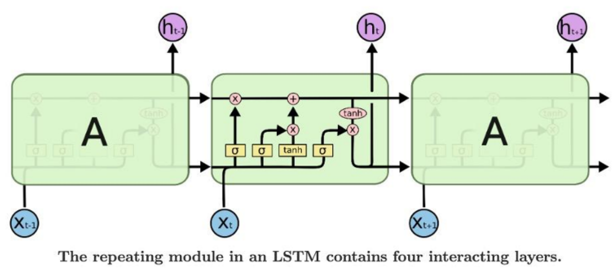

Figure 1. LSTM structure

First, we implemented the traditional LSTM model with a structure similar to the picture above. LSTM, with its gating mechanisms (input gate, forget gate, and output gate), effectively captures long-term dependencies in time series data. However, LSTM faces challenges such as gradient vanishing and low computational efficiency when processing long sequences. To further enhance model performance, we introduced Bidirectional LSTM (BiLSTM) combined with an attention mechanism.

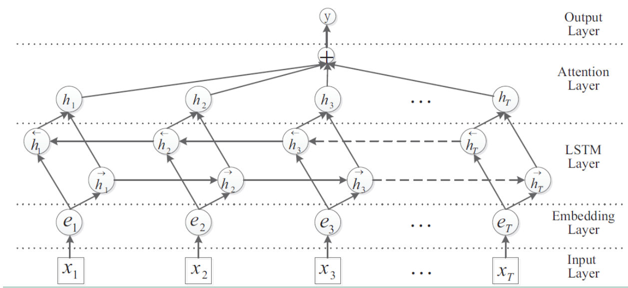

Figure 2. Structure of BiLSTM combined attention

BiLSTM processes time series data in both forward and backward directions, capturing contextual information from both past and future time steps. The attention mechanism dynamically adjusts the importance of different time steps by learning weight vectors, thereby highlighting critical features and suppressing noise.

Next, we proposed the CNN-LSTM hybrid model. This model leverages the strengths of both CNN and LSTM.

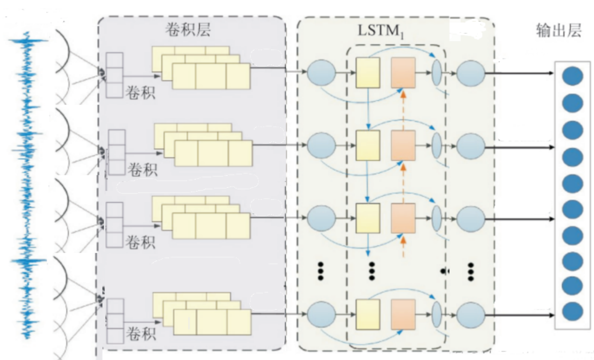

Figure 3. Structure of CNN combined LSTM

CNN extracts local features from traffic volume data through convolutional and pooling layers, while LSTM captures long-term dependencies in the time series. This hybrid architecture simultaneously utilizes spatial and temporal features, thereby improving the model's predictive capability.

Finally, we implemented Temporal Convolutional Networks (TCN). TCN significantly enhances the modeling capability of long sequences through dilated convolutions and multi-layer structures. Dilated convolutions expand the receptive field without increasing the number of parameters, enabling the capture of longer-term dependencies.

To comprehensively evaluate the performance of these models, we adopted Mean Squared Error (MSE) as the loss function and used the Adam optimizer for model training. During training, we introduced dynamic learning rate adjustment strategies and early stopping mechanisms to prevent overfitting. Additionally, we systematically compared the performance of the four models on the training, validation, and test sets, proposing a comprehensive error rate as an evaluation metric to quantitatively compare the strengths and weaknesses of different models.

Our research introduces innovations in the following aspects: First, by incorporating an attention mechanism, we enhanced BiLSTM's ability to focus on critical time steps. Second, by combining the strengths of CNN and LSTM, we proposed the CNN-LSTM hybrid model, which captures both local features and long-term dependencies. Finally, through systematic comparison of LSTM, BiLSTM+Attention, CNN-LSTM, and TCN, we revealed their respective strengths and limitations, providing new perspectives for the field of traffic volume prediction.

4. Experiments

In this chapter, we will provide a detailed introduction to the experimental setup, including the hardware and software environment, the dataset used, the implementation of the models, and the training process.

First, we will describe the experimental environment, including the choice of computing devices. Next, we will introduce the Metro Interstate Traffic Volume dataset, covering its scale, feature dimensions, and preprocessing steps, such as data splitting and normalization.

Following this, we will elaborate on the implementation details of the four models. This includes their network architectures, parameter initialization, and training strategies. Finally, we will present the loss curves during training, model convergence behavior, and performance evaluation results on the test set.

Through this systematic experimental design, we aim to comprehensively validate the effectiveness of each model in traffic volume prediction tasks and provide reliable data support for subsequent analysis.

4.1. Experiment environment

The experimental environment configuration is as follows:

Name	Configuration information
Operating system	Window 11(x64)
Program language	Python 3.12.4
Framework	Anaconda3 + Pytorch2.4.1
CPU	AMD Ryzen 7 5800H
GPU	NVIDIA GeForce RTX 3060
RAM	16G

Table 1. Experiment environment

4.2. Dataset description

The dataset utilized in this study is the Metro Interstate Traffic Volume dataset, sourced from the UCI Machine Learning Repository. This dataset consists of 48205 valid data points, providing a comprehensive record of hourly traffic volumes on the westbound Interstate 94 highway in Minneapolis-St. Paul, spanning from 2012 to 2018. The dataset encompasses a wide array of contextual information, making it particularly suitable for traffic prediction tasks. The key features of the dataset are detailed below:

Traffic Volume

The primary feature of the dataset is the hourly traffic volume, representing the number of vehicles traveling westbound on Interstate 94. It's critical for understanding and predicting traffic patterns.

Holiday Indicator

A binary feature indicating whether a given day is a holiday. This information is essential for capturing the impact of holidays on traffic volumes, as traffic patterns often differ significantly on holidays compared to regular days.

Weather Features

Temperature: The dataset includes the temperature recorded each hour, measured in degrees Celsius. Temperature can influence driving behavior and traffic flow.

Rainfall: The amount of rainfall recorded each hour, measured in millimeters. Rainfall data is crucial for understanding its effect on traffic volumes and road conditions.

Temporal Features

Date and Time: The dataset contains precise date and time information for each record, allowing for the analysis of temporal patterns in traffic volumes.

Day of the Week: The day of the week is also included, as traffic patterns can vary significantly between weekdays and weekends.

	A	B	C	D	E	F	G	H	I
1	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
2	None	288.28	0	0	40	Clouds	scattered clouds	2012/10/2 9:00	5545
3	None	289.36	0	0	75	Clouds	broken clouds	2012/10/2 10:00	4516
4	None	289.58	0	0	90	Clouds	overcast clouds	2012/10/2 11:00	4767
5	None	290.13	0	0	90	Clouds	overcast clouds	2012/10/2 12:00	5026
6	None	291.14	0	0	75	Clouds	broken clouds	2012/10/2 13:00	4918
7	None	291.72	0	0	1	Clear	sky is clear	2012/10/2 14:00	5181
8	None	293.17	0	0	1	Clear	sky is clear	2012/10/2 15:00	5584
9	None	293.86	0	0	1	Clear	sky is clear	2012/10/2 16:00	6015
10	None	294.14	0	0	20	Clouds	few clouds	2012/10/2 17:00	5791

Figure 4. Partial dataset

The Metro Interstate Traffic Volume dataset is particularly suitable for the application of LSTM networks, which require data with a uniform time distribution to effectively capture temporal dependencies. By preprocessing the data to ensure it meets these requirements, we enhance the LSTM model's ability to accurately predict traffic volumes. Additionally, the rich contextual information provided by the dataset enables the development of comprehensive models that take into account various factors influencing traffic flow.

Overall, the rich features, temporal granularity, and large sample size of the dataset make it an ideal choice for evaluating the performance of our model in traffic flow prediction tasks. It provides a realistic and challenging scenario for testing the ability of each model to handle complex real-world data.

4.3. Data preprocessing

To ensure the accuracy and reliability of the model's prediction results, we conducted detailed data preprocessing on the original traffic volume data.

First, we loaded traffic volume data from the original dataset starting from 2016 and parsed the time column into a datetime format. This step ensured the timeliness of the data while reducing potential interference from historical data on model training. By filtering data from 2016 onwards, we focused on more representative recent data to improve the model's prediction accuracy.

	count	...	std
date_time	23084	...	NaN
temp	23084.0	...	12.627143
rain_1h	23084.0	...	0.518361
clouds_all	23084.0	...	38.926916
traffic_volume	23084.0	...	1968.709181

Figure 5. Data description

After loading the data, we checked for and handled missing and duplicate values. By removing duplicates and addressing missing values, we ensured the completeness and consistency of the data. The presence of missing or duplicate values could introduce bias during model training, making this step critical for ensuring data quality.

Using the Z-score method, we identified and smoothed extreme outliers with a moving average. This reduced the impact of anomalies on model performance. Categorical variables were one-hot encoded for numerical processing. The 'snow 1h' column, which contained many zeros and had an irregular distribution, was removed to avoid skewing the model.

The dataset was split into training (70%), validation (20%), and testing (10%) sets. Numerical features were normalized using Z-score to ensure consistent scaling and improve model stability.

To visually observe trends and fluctuations in the data, we plotted time series graphs for numerical features. These graphs helped us identify anomalies, trends, and periodic patterns in the data. Additionally, we used violin plots to visualize the distribution of normalized features, allowing us to check whether the data satisfied the assumption of normal distribution and further validate the rationality of the preprocessing steps.

DLLP
#*****

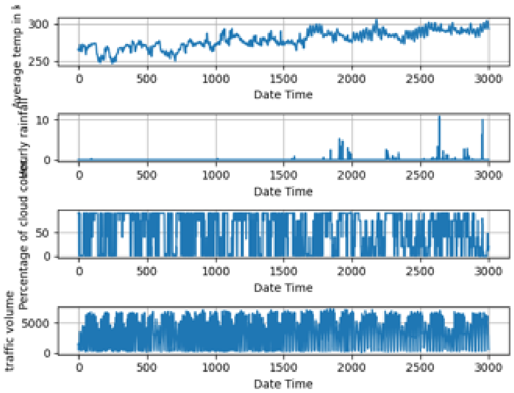

Figure 6. Timeseries line chart

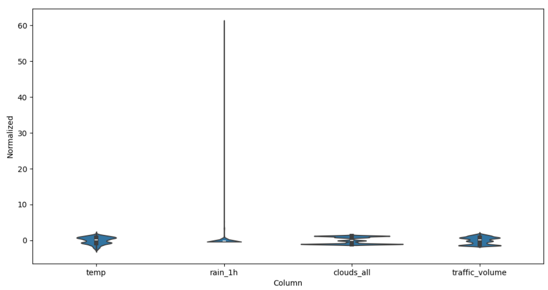

Figure 7. Violin chart

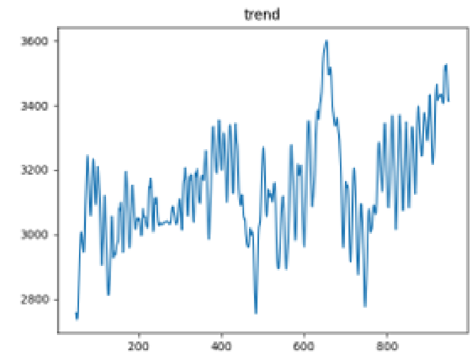

Figure 8. Trend line chart

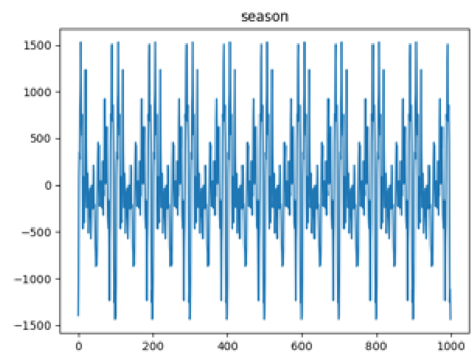

Figure 9. Seasonality line chart

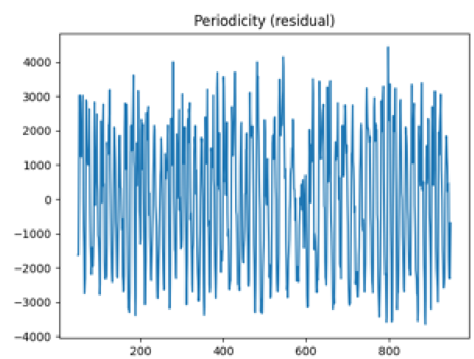

Figure 10. Periodicity line chart

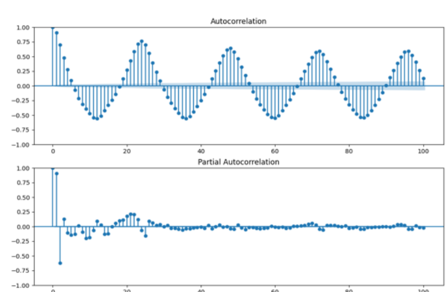

Figure 11. Autocorrelation chart

From the violin plots, we observed the following distribution characteristics of the variables:

Temperature and traffic volume: exhibit distributions that closely resemble a normal distribution, with data primarily concentrated around the mean and fewer values at the extremes.
Cloud cover: shows a distribution with more values at the extremes and fewer in the middle, indicating a more dispersed data distribution.
Rainfall: displays an extreme distribution, where a few outliers result in a large range, but the majority of the data is concentrated within the range of 0 to 5.

To gain a deeper understanding of the characteristics of the traffic volume data, we performed trend, seasonal, and residual analysis using seasonal decomposition. This analysis revealed long-term trends, seasonal fluctuations, and random components in the data. Moreover, we conducted a stationarity analysis on the traffic volume data using the Augmented Dickey-Fuller (ADF) test, which confirmed that the data is stationary and suitable for time series modeling. Finally, we plotted autocorrelation function (ACF) and partial autocorrelation function (PACF) graphs to determine the lag order for the time series model, providing a scientific basis for model selection and parameter optimization.

Based on the analysis results, the ADF test yielded an ADF statistic of **-19.2932**, with a p-value **less than 0.05**, indicating that the time series is stationary. Visualization of seasonal decomposition reveals that the traffic volume data exhibits strong seasonal patterns and cyclicality, suggesting clear periodic fluctuations over time.

The ACF plot shows that the autocorrelation peaks at lag 0 and gradually decays as the lag increases, indicating short-term dependencies in the data. Furthermore, the PACF plot demonstrates significant partial autocorrelation at lags 1 and 2, but it becomes insignificant for lags greater than 3, implying that the optimal lag order for the time series model is approximately 3. These findings collectively indicate that the traffic volume data is stationary, exhibits strong seasonality, and has short-term dependencies with an optimal lag order of around 3, providing a robust basis for selecting and tuning time series models to accurately predict traffic volume.

Through the above data preprocessing steps, we ensured the quality and suitability of the data, laying a solid foundation for subsequent model training and evaluation. These efforts not only improved the model's predictive performance but also provided reliable support for in-depth analysis of traffic volume data.

4.4. Model construction and training

In this project, we constructed four different deep learning models to predict traffic volume data, including LSTM, Bidirectional LSTM with Attention, CNN-LSTM hybrid model, and Temporal Convolutional Network. Below, we provide a detailed explanation of the structure of each model, the training process, and the evaluation methods.

4.4.1. Model architecture

(1) LSTM Model

Input Layer: The input feature dimension is input_size, representing the number of features at each time step.

LSTM Layer: Contains num_layers LSTM layers, with each layer's hidden state size being hidden_layer_size.

Fully Connected Layer: Maps the hidden state of the last time step from the LSTM layer to the output dimension output_size (i.e., the predicted traffic volume).

Forward Propagation: The input data passes through the LSTM layer, and the hidden state of the last time step is used to generate the prediction via the fully connected layer.

(2) Bidirectional LSTM with Attention

Bidirectional LSTM Layer: Contains num_layers bidirectional LSTM layers, with each layer's hidden state size being hidden_layer_size. The output dimension of the bidirectional LSTM is hidden_layer_size * 2.

Attention Mechanism: Computes attention weights for each time step through a fully connected layer and performs a weighted sum on the LSTM outputs to obtain a context vector.

Fully Connected Layer: Maps the context vector to the output dimension output_size.

(3) CNN-LSTM Hybrid Model

1D Convolutional Layer: Performs 1D convolution on the input data to extract local features. The kernel size is 3, and the output channel number is 64.

LSTM Layer: Takes the output of the convolutional layer as input to capture long-term dependencies in the time series.

Fully Connected Layer: Maps the hidden state of the last time step from the LSTM layer to the output dimension output_size.

(4) Temporal Convolutional Network

Temporal Convolutional Blocks: Consists of multiple temporal convolutional blocks, each containing two 1D convolutional layers with dilated convolutions to expand the receptive field.

Residual Connections: Introduces residual connections in each convolutional block to avoid gradient vanishing issues.

Fully Connected Layer: Maps the output of the last time step to the output dimension output_size.

4.4.2. Training

During the model training phase, we adopted a series of optimization strategies to enhance the model's performance.

First, the preprocessed and formatted training data was loaded. During training, we selected Mean Squared Error (MSE) as the loss function to measure the difference between predicted and true values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

N is the number of samples, y_i is the true value for the i-th sample, $\hat{y}_i$ is the predicted value for the i-th sample.

MSE imposes a higher penalty on larger errors, thereby encouraging the model to make more accurate predictions.

Next, we used the Adam optimizer for parameter optimization, with an initial learning rate set to 0.005. The Adam optimizer combines the advantages of adaptive learning rates and momentum, enabling efficient training of deep learning models[9].

To further improve training efficiency, we introduced a dynamic learning rate adjustment strategy (ReduceLROnPlateau). When the validation loss did not decrease for 5 consecutive epochs, the learning rate was reduced to 10% of its original value. This mechanism helped the model escape local optima and converge to better solutions.

During the training process, within each epoch, the model performed forward propagation, loss computation, and backpropagation on the training set. The validation loss was calculated on the validation set to monitor the model's generalization ability. Every 5 epochs, the current training progress was outputted.

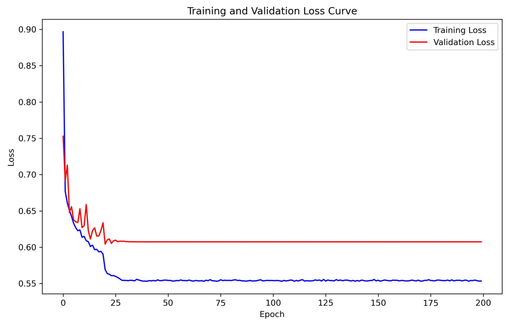

Figure 12. LSTM training and validation loss

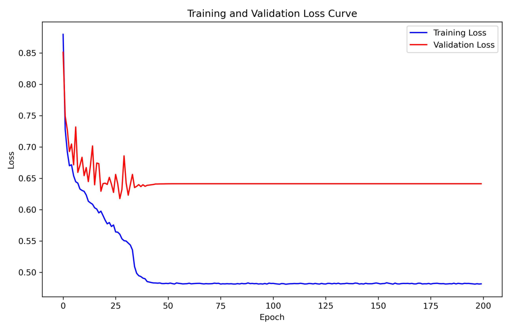

Figure 13. BiLSTM training and validation loss

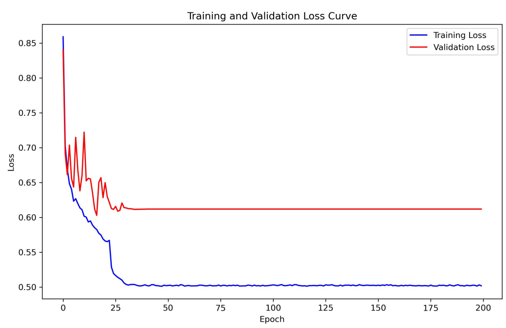

Figure 14. CNN+LSTM training and validation loss

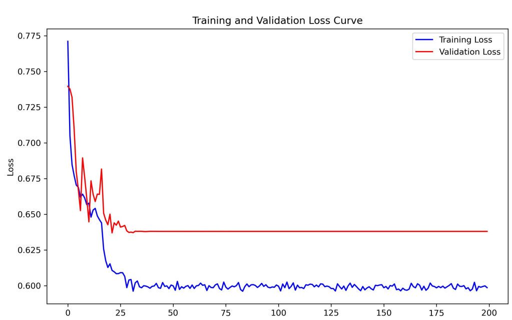

Figure 15. TCN training and validation loss

Model	Train_loss	Validation_loss	Test_loss
LSTM	0.5535	0.6075	0.4759
BiLSTM+Attention	0.4815	0.6414	0.5130
CNN+LSTM	0.5019	0.6119	0.4645
TCN	0.5986	0.6380	0.4525

Table 2. Loss summary

We visualized the predicted traffic volume against the actual data, and the results are as follows:

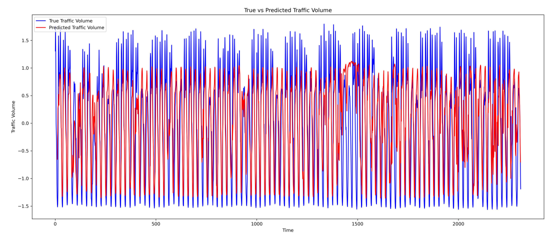

Figure 16. LSTM predicted vs actual

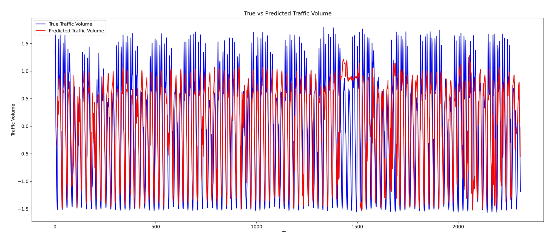

Figure 17. BiLSTM+Attention predicted vs actual

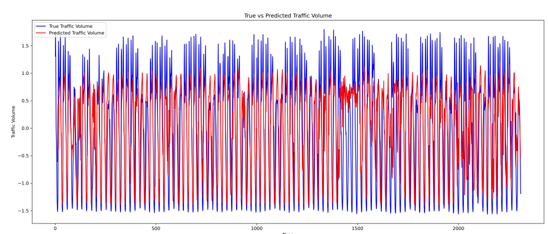

Figure 18. CNN+LSTM predicted vs actual

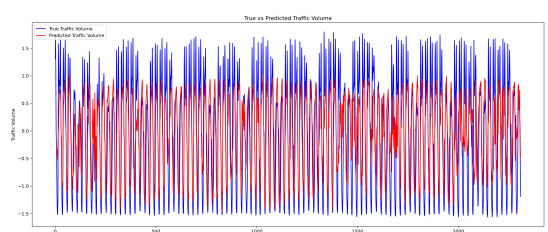

Figure 19. TCN predicted vs actual

In the comparison of the four models, the LSTM model demonstrates stable performance on the test set, but its higher losses on the training and validation sets indicate limited expressive power, potentially failing to fully capture the complex patterns in the data.

The BiLSTM+Attention model excels on the training set but shows higher losses on the validation and test sets, indicating clear overfitting issues, which may require further adjustments to the model structure or the introduction of regularization techniques.

The CNN-LSTM model achieves relatively low losses across the training, validation, and test sets, with performance on the test set close to that of TCN, demonstrating a good balance between fitting capability and generalization ability.

The TCN model exists a slight fluctuation trend after convergenceachieves.It has the lowest loss on the test set, indicating the best performance on unseen data and making it suitable for practical traffic volume prediction tasks, despite its higher losses on the training and validation sets, which suggest some degree of underfitting.

5. Conclusion

This study is based on the Metro Interstate Traffic Volume dataset and compares the performance of four deep learning models—LSTM, BiLSTM+Attention, CNN-LSTM, and TCN—in traffic flow prediction. The experimental results show that the TCN model performs best on the test set, demonstrating excellent generalization capabilities. The CNN-LSTM model exhibits balanced performance across the training, validation, and test sets, effectively balancing fitting and generalization. The BiLSTM+Attention model performs exceptionally well on the training set but shows signs of overfitting on the validation and test sets. The LSTM model demonstrates stable performance on the test set, but its higher losses on the training and validation sets indicate limited expressive power.

Compared to existing research, the innovation of this study lies in the systematic comparison of multiple models and the introduction of attention mechanisms and hybrid architectures such as CNN-LSTM, which significantly improve prediction performance. Unlike traditional studies focusing on single models, our work provides a more comprehensive understanding of the applicability and potential issues of different models. For example, it highlights the advantages of TCN in long-sequence modeling and the challenges of overfitting in BiLSTM+Attention.

The strengths of this study include the use of a real-world, large-scale dataset and the implementation of detailed experimental designs and optimization strategies to ensure the reliability of the results. However, the lack of additional external features in the dataset may limit the models' ability to predict complex scenarios. Furthermore, there are shortcomings in the adjustment of model. Future work could address this issue by increasing regularization or adjusting the attention mechanism's structure.

This study provides an efficient technical solution for traffic flow prediction and, through systematic comparison, reveals the strengths and limitations of each model, offering empirical support for the application of deep learning in time series prediction. Additionally, the multi-model comparison framework and evaluation metrics proposed in this study can be extended to other time series prediction tasks, such as energy consumption forecasting and financial market prediction.

6. Members' contribution

Chen Chen, contribution score: 25: Implementating the basic architecture of LSTM model, Implementing the training and evaluation and visualization process, Writing midterm and final course paper.

Wenjie Liang, contribution score: 25: Data collection and preprocessing, Construction of Data Feature Engineering, Implementating CNN+LSTM and TCN model and BiLSTM+Attention model,Making PowerPoint presentations for the beginning and midterm report.

Peilin Lin, contribution score: 25: Participate in topic selection, Designing the structure of model, Implementating TCN model.

Jingwei Wu, contribution score: 25: Participate in topic selection, Collecting relevant research contents.

References

[1] B. M. Williams and L. A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664-672, Jun. 2003.

[2] B. L. Smith, B. M. Williams, and R. K. Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303-321, Aug. 2012.

[3] Y. Zhang and Y. Xie, Forecasting short-term traffic flow by support vector machine with genetic algorithms, *Expert Systems with Applications*, vol. 34, no. 1, pp. 328-336, Jan. 2008.

[4] T. Chen, C. Guestrin, and X. He, XGBoost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, Aug. 2016.

[5] R. Yu, Y. Li, and C. Shahabi, Deep learning: A generic approach for extreme condition traffic fore-

casting, *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 777-785, Apr. 2017.

[6] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, *arXiv preprint arXiv:1409.1259*, Sep. 2014.

[7] S. Bai, J. Z. Kolter, and V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, *arXiv preprint arXiv:1803.01271*, Mar. 2018.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998-6008, Dec. 2017.

[9] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, Dec. 2014.

1
2
2
2
2
2
2
2
6