DOI: 10.54254/2755-2721/2026.33549

A Comparative Study of RNN, LSTM and GRU for Short-Term Traffic Speed Prediction

Qiuyi Zhao

School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
Qiuyi.Zhao24@student.xjtlu.edu.cn

Abstract. Due to the complex and dynamic time patterns in the actual traffic system, accurate traffic speed prediction is still challenging. This study will focus on using deep learning sequence models to model short-term traffic dynamics. To achieve a fair and controllable comparison, three representative models, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are implemented under a unified framework. The sliding window strategy is used to organize the input data into a multivariate time series, where historical observations of 12 time steps are used to predict future traffic speeds. All models share the same architecture design, which is composed of a stacked loop layer and a fully connected layer, ensuring the consistency of model capacity and training conditions. The training process is optimized by using the Adam algorithm with the mean square error as the loss function. Based on the comparative results, the best performing model was further analyzed through a series of controlled experiments. The system changes key hyperparameters, including hidden layer size, network depth and learning rate, to check their impact on prediction performance. This design allows the impact of model capacity and optimization dynamics to be evaluated in isolation. The experimental results show that under the short-term prediction setting, the simpler loop structure can obtain competitive or even superior performance compared with the more complex gating model.

Keywords: traffic prediction, time series forecasting, RNN, LSTM, GRU

1. Introduction

In recent years, with the acceleration of urbanization, the problem of traffic congestion has become increasingly serious, which has seriously affected the efficiency of urban transportation systems. For example, it will increase people 's commuting time to work. Therefore, an accurate traffic prediction model can help people identify potential congestion in advance, so as to avoid potential congestion, which is of great significance today.

With the increasing number of traffic sensors, it is very easy to collect traffic data and use its training model to predict future traffic conditions. Although traditional statistical methods such as ARIMA are common in time series analysis, they are difficult to capture the complex and nonlinear time dependence inherent in traffic flow [1]. The deep learning model is good at extracting sequential patterns and achieves excellent results in predictive applications [2,3]. Recurrent Neural

DOI: 10.54254/2755-2721/2026.33549

Networks (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are particularly popular because they can learn time dynamics from sequence data [4].

Although they all have powerful functions, it is still challenging to choose the best architecture. Because of the lack of systematic comparison between these three frameworks, the model selection process becomes more complicated. In addition, recent literature often prioritizes structural novelty over basic hyperparameters - such as network depth, learning rate, or hidden layer size - thus limiting actual optimization efforts. Many evaluations also rely heavily on a single performance indicator, rather than a complete set of indicators.

To address this gap, this study used multiple error metrics, including MAE, RMSE, MSE, and R2 coefficients, to compare the RNN, LSTM, and GRU frameworks. On the basis of this evaluation, this study further separated the best performing models to study the impact of key hyperparameter adjustments on the prediction results, aiming to establish a strict experimental baseline for traffic volume prediction.

2. Method

2.1. Dataset preparation

The traffic speed data set is derived from the PeMS data released by the DCRNN project [5]. The dataset contains multivariate time series observations collected from 207 traffic sensors. The traffic prediction task is formalized as a multivariable regression problem. A sliding window strategy is adopted to construct supervised learning samples. The traffic observation values of the first 12 time steps (representing a 60-minute historical window at a 5-minute interval) are used to predict the traffic speed of the next time step.

To enhance the numerical stability and accelerate the convergence of the model, the input data is standardized by Z-score. Normalization parameters, including mean and standard deviation, are calculated only on the training set and then applied to the validation set and test set to prevent data leakage. Finally, the dataset is divided into training (70 %), validation (10 %) and test (20 %) subsets in chronological order.

2.2. Deep learning models

To ensure strict and fair comparison, all three benchmark models (RNN, LSTM and GRU) are implemented using the same architecture framework. The input data is structured as a three-dimensional tensor, including batch size, sequence length (12 time steps), and number of sensors (207). Each model processes continuous traffic observations to extract hidden representations. The output of the hidden state of the last time step is extracted and mapped through a shared fully connected linear layer, the representation of which can predict the traffic speed of all sensors at the same time. In the baseline of this study, all models are composed of two loop layers with a hidden state size of 64.

2.2.1. Recurrent neural network (RNN)

Standard RNN is the basic model of sequence analysis. It uses cyclic connections to transfer hidden states across time steps, so that the network can maintain a memory of recent traffic conditions. Compared with the gated variant, RNN has a simpler structure and fewer parameters, resulting in lower computational overhead. Due to the gradient disappearance problem, standard RNN is usually difficult to deal with long-term dependencies. The sliding window is limited to 12 time steps. In this

DOI: 10.54254/2755-2721/2026.33549

localized window, RNN is theoretically still able to capture real-time, short-term traffic fluctuations [6].

2.2.2. Long short-term memory (LSTM)

In order to overcome the limitations of the standard RNN in preserving the extended historical context, LSTM introduces an explicit memory unit, which is regulated by three mechanisms : input, forgetting, and output gates [7]. In the traffic prediction problem, the forgetting gate determines which old traffic patterns are discarded, and the input gate determines which new observations are related to the updated cell state. By controlling the information flow, LSTM effectively alleviates the problem of gradient disappearance, so that it can reflect the sudden traffic anomalies and continuous congestion trends over time.

2.2.3. Gated recurrent unit (GRU)

GRU 's principle is to capture time dependencies similar to LSTM, but it has a more streamlined architecture. It simplifies memory control into two parts: update gate and reset gate. The update gate controls the preservation of past traffic status (integrating the functions of LSTM 's forgotten gate and input gate), while the reset gate controls the influence of past status on current candidate status. Compared with LSTM, this design allows GRU to achieve robust modeling of nonlinear traffic dynamics while reducing memory consumption and accelerating training [8, 9].

2.3. Hyperparameter configuration

All models in this paper are implemented in Python using the PyTorch framework. The length of the input sequence is fixed to 12, and the batch size is set to 64. Model training is optimized by Adam algorithm. The initial learning rate is 0.001 and a total of 20 epochs are run. Mean square error (MSE) is used as the loss function. Four standard evaluation indexes were used : mean absolute error (MAE), root mean square error (RMSE), mean square error (MSE) and coefficient of determination (R2) [10].

2.4. Hyperparameter study of RNN

In order to study the influence of key hyperparameters on the RNN model with the best performance, a control experiment strategy is adopted, which only changes one hyperparameter at a time, while other hyperparameters remain unchanged. The size of the hidden layer is tested, and the values are 32,64,128. The network depth was tested using 1,2, and 3 loop layers, respectively. The learning rates were assessed as 0.01, 0.001 and 0.0001.

All other training configurations are consistent with the baseline settings described in Section 2.3. This system study aims to reveal how model capacity and optimization dynamics affect short-term traffic prediction performance.

DOI: 10.54254/2755-2721/2026.33549

3. Results and discussion

3.1. Result

3.1.1. Comparison of sequences models

Table 1 presents the performance comparison among the RNN, LSTM, and GRU models on the traffic speed prediction task. Among them, the RNN model obtains the lowest RMSE(9.321) and MSE(86.878), with the highest coefficient of determination(0.833), indicating better performance in traffic prediction accuracy and stability. The GRU model achieves the lowest MAE(5.356), which means better performance on average prediction error. In contrast, the LSTM model show s relatively higher error values across most metrics.

As shown in Figure 1, all models can capture the general temporal trend of traffic speed variations, including the sudden fluctuations. Figure 2 shows that all models demonstrate stable convergence during training. The RNN converges to a lower validation loss compared with the other models, indicating more stable optimization behaviour.

Table 1. Performance comparison

Model	MAE	RMSE	MSE	R^2
RNN	5.416	9.321	86.878	0.833
LSTM	5.372	9.609	92.335	0.822
GRU	5.356	9.369	87.775	0.831

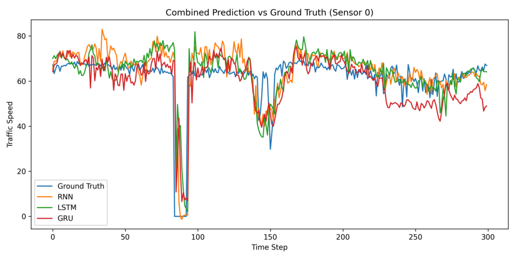

Figure 1. Prediction comparison (picture credit: original)

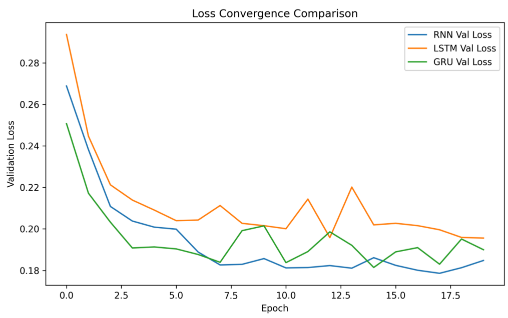

Figure 2. Loss convergence (picture credit: original)

3.1.2. Result of hyperparameter study on RNN

To investigate the impact of key hyperparameters on the prediction performance of the RNN model, a series of controlled experiments were conducted by varying one parameter at a time while keeping

DOI: 10.54254/2755-2721/2026.33549

others fixed. The experimental results are summarized in Tables 2–4.

Table 2 shows the increasing the size of the hidden layer generally improves the prediction accuracy. When the hidden size increases from 32 to 128, the prediction error decreases and the R2 value increases, indicating that a deeper hidden size helps the model to better capture short-term traffic dynamics.

Table 3 shows the impact of network depth. The results show that the single-layer RNN achieves the best performance, while the deeper structure leads to a slightly higher prediction error. This shows that under the short input sequence used in this study, increasing the depth of the model does not bring additional benefits.

The impact of learning rate is summarized in Table 4. Among the test values, the learning rate of 0.001 provides the most stable and accurate prediction results, while both larger and smaller learning rates lead to performance degradation.

Overall, these results indicate that appropriate hyperparameter settings play an important role in improving RNN prediction performance.

Table 2. Effect of hidden size(network depth = 2, learning rate = 0.001)

Hidden size	MAE	RMSE	MSE	R2
32	5.71	10.16	103.18	0.80
64	5.41	9.32	86.87	0.83
128	5.15	8.83	77.98	0.85

Table 3. Effect of network depth(hidden size = 64, learning rate = 0.001)

Layers	MAE	RMSE	MSE	R2
1	4.99	8.87	78.65	0.85
2	5.41	9.32	86.87	0.83
3	5.58	9.88	97.69	0.81

Table 4. Effect of learning rate(Hidden size = 64, network depth = 2)

Learning rate	MAE	RMSE	MSE	R2
0.01	5.77	10.40	108.15	0.79
0.001	5.41	9.32	86.87	0.83
0.0001	5.99	10.27	105.53	0.80

3.2. Discussion

3.2.1. Discussion on model comparison

The experimental results show that under the current experimental setting, the three sequence models can capture the overall trend of traffic flow speed changes. The standard RNN achieved slightly better performance in most evaluation indicators, especially the lowest RMSE (9.321) and MSE (86.878). One possible reason is related to the nature of short-term traffic forecasting tasks. In this research, the length of the input sequence is limited to 12 time steps, corresponding to about 1 hour of historical observations. Under this condition, the traffic dynamics are mainly affected by the recent state, and the time dependence range is short.

Therefore, the simple loop structure of RNN seems to be sufficient to model these local temporal patterns. In contrast, LSTM and GRU designs deal with long-term dependencies through gating mechanisms, introducing optimized redundancy for short sequences. The large number of parameters may also make the model training slightly difficult, resulting in unstable performance. This may explain why RNN obtained lower RMSE and MSE values in this study.

At the same time, it can be seen that GRU has achieved the lowest mean absolute error (5.356) in the three models, indicating that it performs better in terms of average prediction error. This shows that different models may have different advantages according to different evaluation perspectives, highlighting the importance of using multiple metrics to evaluate model performance.

3.2.2. Discussion on RNN hyperparameter study

Hyperparametric experiments further show that the configuration of model capacity and training settings has a significant impact on RNN prediction performance. As the size of the hidden layer increases from 32 to 128, the prediction error gradually decreases and reaches a peak when R2 is 0.85 (Table 2). This shows that a larger hidden state dimension allows the model to learn richer time representations from multivariate traffic data. However, increasing the model size may also lead to higher computational costs and greater overfitting risks, which should be considered in practical applications.

In terms of network depth, experimental results show that single-layer RNN achieves better performance than deeper configurations. For short-term prediction tasks with limited time dependence, stacking multiple loops may not provide additional benefits and may even make model optimization more difficult. This finding means that in this case, a relatively simple model structure may be more appropriate.

As for the learning rate, the results show that the moderate value of 0.001 provides a good balance between convergence speed and training stability. A large learning rate may lead to unstable updates during training, while a small learning rate may slow down convergence within a limited number of training steps. These observations indicate that careful tuning of the optimization parameters is critical to achieving stable predictive performance.

4. Conclusion

Under the unified experimental framework, this study systematically compares the short-term traffic prediction of RNN, LSTM and GRU. It can be found that RNN achieves slightly better overall performance while maintaining stable convergence behavior, especially in terms of RMSE and MSE. This experiment also shows that increasing the hidden size can improve the prediction accuracy, and the deeper network structure does not bring additional benefits. In the future, the further study plans to extend this study by incorporating spatial dependencies and exploring more advanced models to further enhance predictive performance.

References

[1] Box, G. E. P., & Jenkins, G. M. (1976).Time series analysis: Forecasting and control. Holden-Day.
[2] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019).Deep learning for time series classification: A review.Data Mining and Knowledge Discovery, 33(4), 917–963.

DOI: 10.54254/2755-2721/2026.33549

[3] Lim, B., Zohren, S., & Roberts, S. (2021).Time-series forecasting with deep learning: A survey.Philosophical Transactions of the Royal Society A, 379(2194).

[4] Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (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] GitHub. (2019). https: //github.com/liyaguang/DCRNN

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

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[8] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1724–1734). Association for Computational Linguistics.

[9] Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 324–328). IEEE.

[10] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research, 30(1), 79–82.