URSI AP-RASC, Sydney, Australia, 17 – 22 August 2025

Numerical modeling of tropospheric atmosphere based on GCN-GRU

Jiangting Li[(1)], Yuxuan Wang* [(1)], Yi Yan[(1)], Zhouxiang Yu[(1)], Zhike Tian[(1)], Kejie Zhang[(1)]
(1) Xidian University, Xi 'an, Shaanxi., China, e-mail: jtli@xidian.edu.cn; 25201110143@xidian.edu.cn
24201111396@stu.xidian.edu.cn; 23201214937@stu.xidian.edu.cn; 24201214698@stu.xidian.edu.cn;
24201214692@stu.xidian.edu.cn

Abstract

This paper presents a novel approach for numerical modeling of the tropospheric atmosphere using a Graph Convolutional Network combined with Gated Recurrent Units (GCN-GRU). The GCN-GRU model integrates spatial and temporal correlations in atmospheric data to enhance the accuracy of tropospheric waveguide prediction. By leveraging the strengths of GCN in capturing spatial dependencies and GRU in handling temporal sequences, this hybrid model demonstrates significant improvements over traditional methods such as the Weather Research and Forecasting (WRF) model. The study focuses on the application of GCN-GRU to predict atmospheric waveguide events over the East China Sea and South China Sea, achieving a prediction accuracy improvement of over 13.56% compared to conventional models. The results highlight the potential of GCN-GRU for high-precision atmospheric numerical prediction and waveguide discrimination.

1. Introduction

The troposphere, the lowest layer of the Earth's atmosphere, is characterized by complex and dynamic meteorological phenomena that significantly impact electromagnetic wave propagation. These phenomena, including atmospheric waveguides, can affect radar detection, communication systems, and navigation accuracy. Accurate prediction of tropospheric waveguide events is crucial for improving the reliability of these systems. This paper introduces a novel approach using a Graph Convolutional Network combined with Gated Recurrent Units (GCN-GRU) to enhance the accuracy and reliability of tropospheric waveguide prediction.

2. GCN-GRU Model

The GCN-GRU model integrates two powerful neural network architectures to capture both spatial and temporal dependencies in atmospheric data. The Graph Convolutional Network (GCN) is designed to handle spatial correlations by modeling the complex topological relationships between different meteorological stations. Each station is treated as a node in a graph, and the edges represent spatial connections based on geographical distances or other similarity metrics. The GCN captures spatial dependencies through convolution operations on the graph structure, allowing the model to aggregate information from neighboring nodes and learn higher-order spatial features.

The Gated Recurrent Unit (GRU) is employed to handle temporal dependencies in the data. GRU is a type of recurrent neural network (RNN) that efficiently captures long-term dependencies in time series data. It uses gating mechanisms to control the flow of information, preventing issues like vanishing gradients that are common in traditional RNNs. By combining GCN and GRU, the model can effectively capture both spatial and temporal correlations in atmospheric data, leading to more accurate predictions of tropospheric waveguide events.

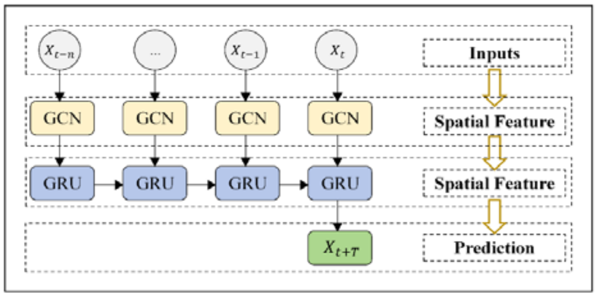


Figure 1. GCN-GRU model calculation process, input first through the graph convolution to capture the spatial correlation, and then input into the GRU unit to capture the time correlation.

3. Data Preprocessing

The data used in this study include meteorological observations from the Integrated Global Radiosonde Archive (IGRA) and numerical simulations from the WRF model.

The data cover a period from 2017 to 2022, focusing on the East China Sea and South China Sea regions (All data is open data). The preprocessing steps involve data cleaning, normalization, and the construction of a spatial graph. Missing values in the IGRA data are handled using linear interpolation, ensuring data continuity.

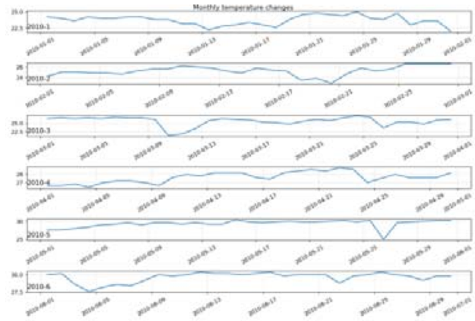


Figure 2. Temperature distribution from January 2010 to June 2010

4. Model Training and Evaluation

The GCN-GRU model is trained using a dataset split into training, validation, and testing sets. The training process involves optimizing the model parameters to minimize the mean squared error (MSE) between predicted and actual atmospheric values. The model's performance is evaluated using metrics such as MSE, root mean squared error (RMSE), and mean absolute error (MAE). The training is performed using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. The model is trained for 50 epochs, with early stopping to prevent overfitting.

5. Results and Discussion

5.1 Variable Correlation Analysis

In the correlation analysis, there is a moderately positive correlation (0.359) between steam and temperature, indicating that an increase in steam is often accompanied by an increase in temperature. On the contrary, there is a strong negative correlation between pressure and temperature (-0.782), indicating that an increase in pressure may lead to a decrease in temperature. The weak correlation between time and other variables indicates that there is no obvious linear trend in the change of steam volume, pressure, and temperature over time.

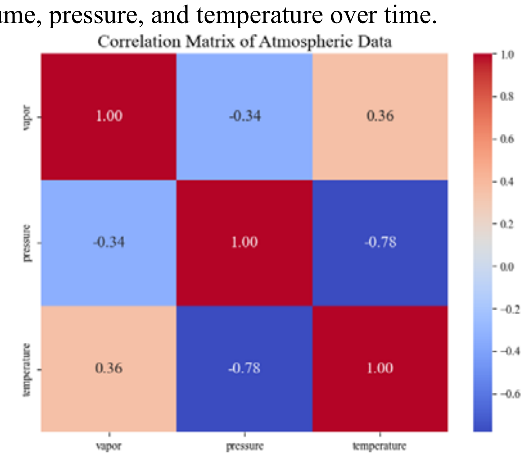


Figure 3. Correlation chart of variables, sorting out correlation between variables within a reasonable range

5.2 Hyperparameter Selection

In the experiments of GCN-GRU model, the main hyperparameters adjusted include learning rate, batch size, training cycle and number of hidden layers. First manually set the learning rate to 0.001, the batch size to 64, and the training period to 50. In addition, the number of hidden units is a very important parameter for the GCN-GRU model, as different numbers of hidden units may significantly affect the prediction accuracy. The optimal combination of hyperparameters is determined by the grid cable.

In the experiment, the IGRA and WRF fusion datasets were used to double-stack the number of hidden units selected from [128, 256, 512, 1024], and the change in prediction accuracy was analyzed. By comparing multiple error metrics, including mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE), and coefficient of determination (R2_Score), you can gain insight into how the model performs in different configurations.

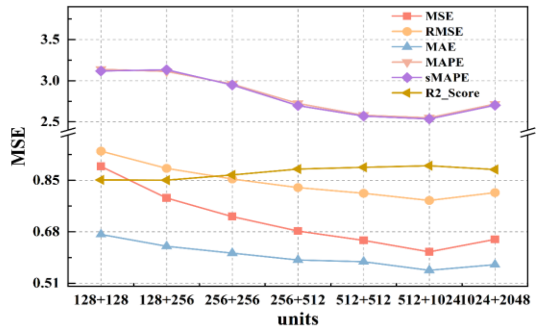


Figure 4. As the stacking number of different memory units increases, the model error decreases first and then increases.

It can be seen from the results that the overall prediction performance of the model improves with the increase of the number of GRU cells. Specifically, when the GRU unit was increased from 128+128 to 512+1024, the MSE decreased from 0.8958 to 0.61363, showing a significant error reduction. Similarly, RMSE and MAE were also reduced from 0.94647 and 0.6713 to 0.78335 and 0.55269, respectively, which further demonstrated the reduction in error. In addition, MAPE and sMAPE showed a similar downward trend, from 3.13914 and 3.11836 to 2.5497 and 2.53519, indicating that the accuracy and reliability of the forecasts were enhanced.

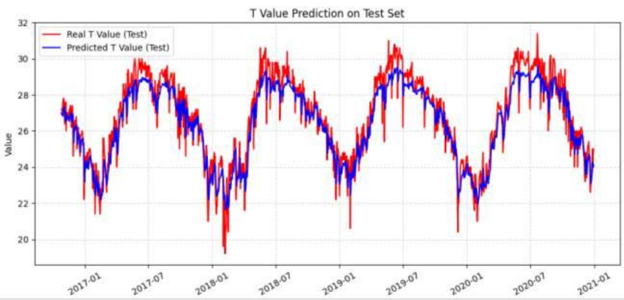
It is worth noting that the value of R2_Score, as a measure of the model's ability to explain variables, also increases with the number of units. In the 128+128 configuration, the R2_Score was 0.85128, while in the 512+1024 configuration, the value increased to 0.89816, showing that the model's ability to explain data variability was enhanced.

However, in the 1024+2048 configuration, MSE, RMSE, and MAE increased compared to the 512+1024 configuration to 0.65496, 0.80929, and 0.57135, respectively, while R2_Score also decreased to 0.88578. This may indicate that the model begins to overfit under this configuration, that is, the model is too sensitive to the training data, resulting in reduced generalization.

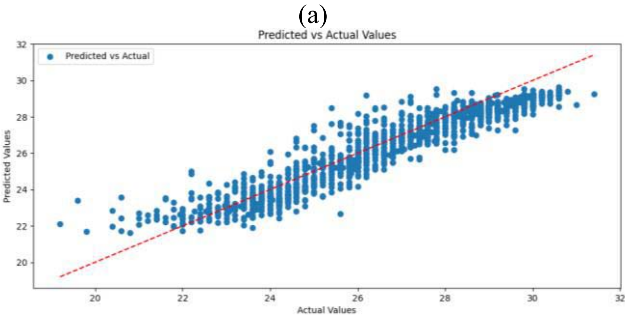
In summary, through the detailed comparison of different GRU unit combinations, it can be concluded that increasing the number of GRU units can effectively improve the prediction performance of the model, but it is also necessary to pay attention to the appropriate number of units to avoid overfitting problems. When the model GRU units are stacked to 512+512, the overall calculation time and accuracy are at a high level. The stacking mode of 512+512 was used in the subsequent experiments.

5.3 Result Analysis

After the number of memory units is selected as 512+512 stack, the overall prediction results and scatter plot results of the model are shown in the figure below.



(a)



(b)

Figure 5. Model prediction results: (a) Prediction result graph (b) prediction scatter graph

GRU model and sGRU model were selected as control experiments, and the error results of the comprehensive prediction were shown in the following figure.

Table1. Comparison of error results

	GRU	sGRU	GCN-GRU
MSE	0.7432	0.7189	**0.7044**
RMSE	0.8621	0.8479	**0.8392**
MAE	0.6929	0.6424	**0.6284**
R^2	0.8768	0.8813	**0.8845**

The results showed that the GCN-GRU model had the best performance, with MSE of 0.70442, RMSE of 0.839297, MAE of 0.628406 and R^2 of 0.884527. This indicates that a properly increased stack of memory units can significantly improve model performance, reduce errors and improve goodness of fit. More parameters allow the model to capture more complex data patterns, improving the overall prediction accuracy. In conclusion, the prediction performance of GCN-GRU model can be significantly improved by setting the number of memory units reasonably

References

[1] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.

[2] R. Wan, S. Mei, J. Wang, M. Liu, F. Yang, Multivariate temporal con- volutional network: A deep neural networks approach for multivariate time series forecasting, Electronics 8 (2019) 876.

[3] S. Ma, T. Zhang, Y.-B. Zhao, Y. Kang, P. Bai, Tcln: A transformer- based conv-lstm network for multivariate time series forecasting, Applied Intelligence 53 (2023) 28401–28417.

[4] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, R. Salakhutdinov, Transformer-xl: Attentive language models beyond a fixed-length con- text, arXiv preprint arXiv:1901.02860 (2019).

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[6] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, In- former: Beyond efficient transformer for long sequence time-series fore- casting, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 11106–11115.

[7] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al., Time-llm: Time series forecasting by reprogramming large language models, arXiv preprint arXiv:2310.01728 (2023).