

StemGNN for Divvy Bike Re-balancing

Noah Anderson

Introduction

Methods

In this section, the methodology employed in forecasting bicycle shortages at various Divvy Stations in Chicago using a modified version of the StemGNN model originally developed by Microsoft (StemGNN repository, Microsoft, 2023) will be described. The dataset used for this study was sourced from the Chicago Data Portal (City of Chicago, 2023), covering the period from March to September 2022. This dataset provided comprehensive historical information regarding the status of Divvy stations, including the availability and total number of docks at each station, recorded at 10-minute intervals. To facilitate model tuning and exploratory analysis, a focus was placed on specific neighborhoods, namely Lake View, Lincoln Park, Uptown, and North Center, which collectively encompassed 49 bike stations. Unfortunately, data for 8 stations were unavailable due to a download error. However, the data from the remaining 41 stations was deemed sufficient for analysis. Subsequently, the data was reformatted into a T*N matrix, with ‘T’ representing time stamps and ‘N’ representing nodes. Each entry in this matrix reflected the bicycle deficit at a specific node and time.

The StemGNN model, originally proposed by Cao et al. (2020), incorporates Discrete Fourier Transform (DFT) and Graph Fourier Transform (GFT) methodologies. DFT is employed to model temporal dependencies, enabling the identification of patterns such as seasonality and autocorrelation. On the other hand, GFT is utilized to analyze interseries correlations by examining spatial interactions between nodes.

In this research, the focus was on fine-tuning two primary parameters of the StemGNN model: the learning rate and window size, while maintaining the default settings for other parameters. The learning rate influences the model’s learning speed, and the window size specifies the quantity of past data points used for forecasting. Experiments were conducted with forecast horizons of 3 and 6 (equivalent to 30 and 60 minutes into the future), learning rates of 0.01, 0.001, and 0.0001, and window sizes of 6, 12, 24, and 30, resulting in a total of 24 distinct model configurations.

Root Mean Square Error (RMSE) served as the primary evaluation metric to assess model performance. It is worth noting that RMSE can be calculated in various ways for a joint time series, as it effectively consists of different models, one for each node. In this study, RMSE for the entire output, encompassing both time and space, was computed as the chosen metric for evaluation. This approach provides a comprehensive measure of model performance across both temporal and spatial dimensions.

Results

Optimal Model Configuration

The superior performance for both 30-minute (horizon 3) and 60-minute (horizon 6) forecasts was observed with a learning rate of 10^{-3} and a window size of 12. This configuration yielded an RMSE of 0.796 docks for horizon 3 and 0.813 docks for horizon 6.

Table 1: Table 1: Detailed RMSE Results for the 30-Minute Forecast

Horizon	LearningRate	WindowSize	RMSE
3	1e-04	6	2.6609161
3	1e-04	12	0.9441134
3	1e-04	24	0.9691645
3	1e-04	30	0.9570124
3	1e-03	6	0.8248647
3	1e-03	12	0.7956052
3	1e-03	24	0.8157663
3	1e-03	30	0.8720042
3	1e-02	6	0.9944132
3	1e-02	12	1.8608359
3	1e-02	24	2.4697193
3	1e-02	30	2.2420155

Table 2: Table 2: Detailed RMSE Results for the 60-Minute Forecast

Horizon	LearningRate	WindowSize	RMSE
6	1e-04	6	1.1587956
6	1e-04	12	0.9617886
6	1e-04	24	0.9950640
6	1e-04	30	1.0493857
6	1e-03	6	0.8173219
6	1e-03	12	0.8129242
6	1e-03	24	0.8553031
6	1e-03	30	0.9477599
6	1e-02	6	1.4051342
6	1e-02	12	2.1851017
6	1e-02	24	2.4220015
6	1e-02	30	2.7565324

Impact of Tuning Parameters on Performance

The study highlighted the pivotal role of learning rate as a critical tuning parameter. A learning rate of 10^{-2} was too rapid, leading to less accurate models, whereas 10^{-4} , though slower, was effective with larger window sizes. The rate of 10^{-3} emerged as optimal across different window sizes, with its performance slightly diminishing beyond a window size of 12. These effects can be seen in Figures 1 and 2. Notably, smaller window sizes, which are quicker to train, proved as effective as larger ones, offering a balance between accuracy and efficiency. For full model results see Tables 1 and 2.

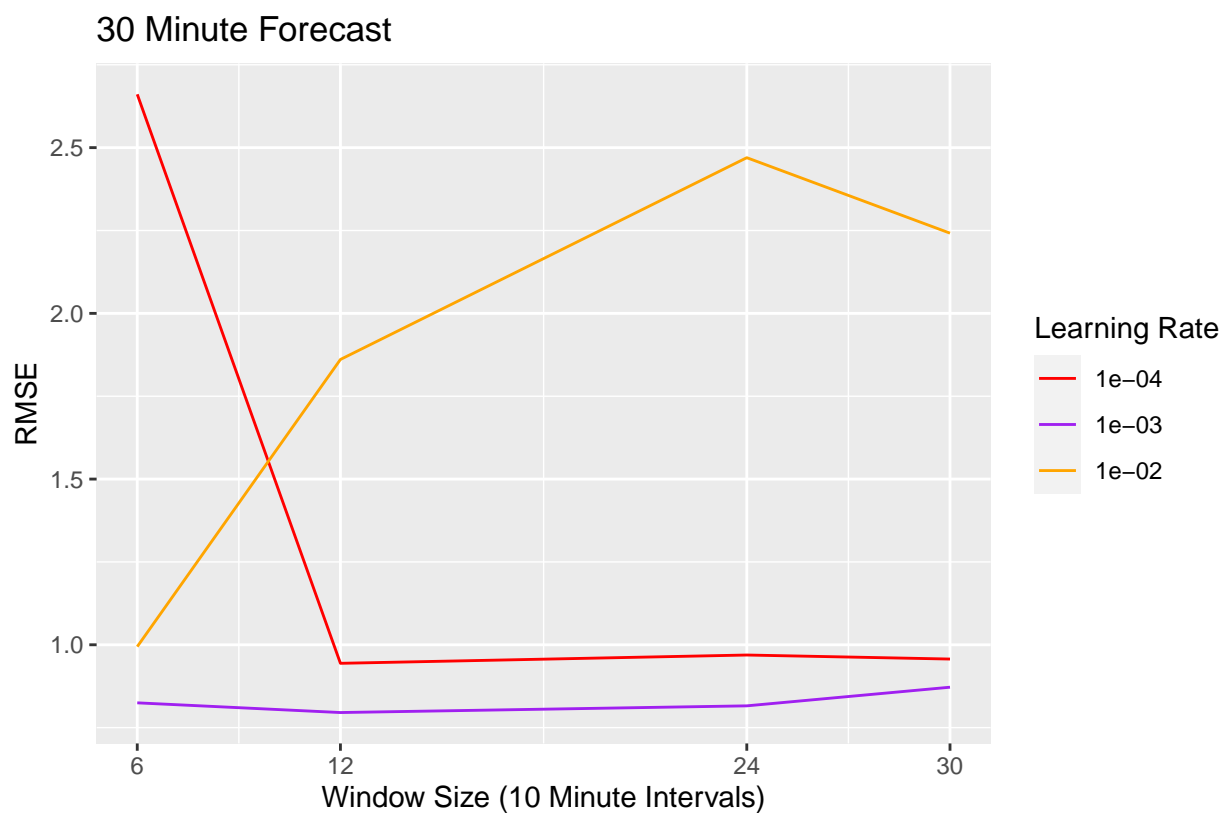


Figure 1: RMSE across different window sizes for the 30-minute forecast.

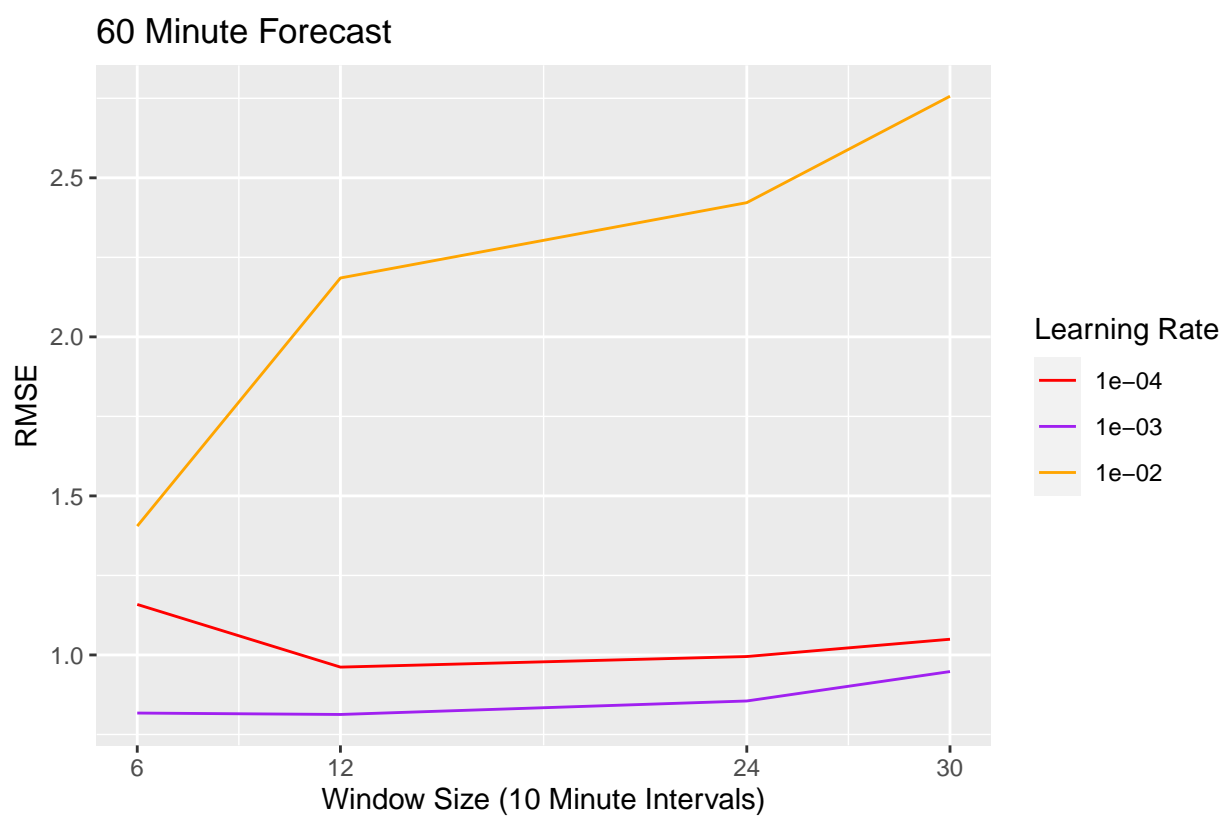


Figure 2: RMSE across different window sizes for the 60-minute forecast.

Discussion

Efficiency versus Accuracy

When examining the learning rate of 10^{-3} , the difference in RMSE between window sizes 6 and 12 was minimal (0.03 docks for horizon 3 and 0.004 docks for horizon 6). This indicates that the marginal improvement with a window size of 12 does not justify its longer training time. Consequently, satisfactory results can be attained with fewer resources, an important consideration for practical model deployment.

References

Microsoft. (2023). StemGNN: Spectral Temporal Graph Neural Network. GitHub. Retrieved from <https://github.com/microsoft/StemGNN>

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., . . . & Zhang, Q. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33, 17766-17778.

City of Chicago. "Divy Trips." Data published by City of Chicago. Accessed on November 27, 2023. Available at: <https://data.cityofchicago.org/Transportation/Divy-Trips/fg6s-gzvg>.

Appendix