2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

# Train arrival delay prediction based on spatial-temporal graph convolutional network to sequence model*

Jianmin Li, Xinyue Xu, Rui Shi, and Xin Ding

*Abstract*— **This paper proposes a spatial-temporal graph convolutional network (GCN) to sequence model (STG2Seq) to predict multi-step train arrival delays. First, the train delay data is preprocessed and encoded as the input of GCN layers. Second, gated graph convolutional module (GGCM) is composed of several GCN layers, and then GGCM is stacked to build long-term encoder and short-term encoder respectively. Among them, long-term encoder is designed to encode historical train delay data, and short-term encoder is introduced to derive the next-step prediction for generating multi-step prediction. Third, an attention-based output module is constructed to splice the output of the long-term and short-term encoder, and then important information is extracted based on channel-wise attention. Finally, the performance of proposed method for train delay prediction is evaluated based on train operation records from Wuhan-Guangzhou high-speed railway. The numerical results show that proposed method outperforms other baseline models.**

## I. Introduction

Train delay prediction is an important part of railway delay management and the key to the optimization of timetable rescheduling. In recent years, with the rapid development of Chinese High-speed Railways (HSR), breakthroughs have been made in both speed and mileage. However, the high-speed train will inevitably be affected by some abnormal events. For one thing, they will seriously affect railway transportation organizations and reduce the quality of railway transportation services. For another thing, they will seriously affect passengers' travel plans and increases travel time [1]. Therefore, how to accurately predict the train delay is a new challenge for railway operation management. In addition, the accurate estimation of train delay is helpful to the reasonable formulation of train timetable, and provides decision-making basis for the train dispatcher [2].

Train delay forecast is a process based on delay probability estimation, which has been widely concerned by researchers [2]–[5]. The traditional statistical model is initially used in the problem of train delay prediction, such as linear regression and multiple regression models [6]. Furthermore, some delay probability distribution models were proposed to estimate train delay [7]. For example, an exponential distribution model was proposed, which applied a three-way, two block station train delay propagation signal system, to estimate delays of trains caused by train operational accidents [8]. However, the traditional prediction model lacks the ability to learn the hidden knowledge of train operation data [9].

Fortunately, with the rapid development of machine learning models, increasingly machine learning models are used to predict train delays, which make up for the shortcomings of traditional methods. ANN (Artificial neural network) is a popular method of researchers for delay prediction [10], [11]. Masoud et al. [11] used ANNs to predict the train delay with three different models, including standard real number, binary coding and binary set encoding inputs. However, the prediction accuracy of ANNs is not sufficient, along with the extraneous parameter adjustment process. Further, Marković et al. [1] used support vector regression to predict train delay, in order to capture the relationship between train delay and various external factors. The results show that support vector regression is better than ANN. Other machine learning methods for train delay prediction include Extreme Gradient Boosting (XGBoost) [12], Bayesian network [5], [13], [14], extreme learning machine [15], [16] and so on. Shi et al. [12] proposed a hybrid method that combines XGBoost and Bayesian optimization algorithm to predict train arrival delays. Li et al. [15] used extreme learning machine tuned via particle swarm optimization to predict train arrival delays. In general, machine learning method has fewer model assumptions than traditional statistical methods, generally can get better data fitting results, and has better adaptability, so it has been successfully applied in the field of train delay prediction.

In brief, after summarizing a large number of excellent research on train delay prediction, we find that most of the current research focus on one-step train delay prediction, while the researches on multi-step train delay prediction are insufficient. At present, Huang et al. [9] is one of the few classic works on multi-step delay prediction. Huang et al. proposed a hybrid method, which combines two LSTM models and a fully connected neural network to predict train delay. Furthermore, Huang et al. [3] proposed a deep learning method to predict train delay. However, this kind of method relies heavily on the structure of RNN-base to capture temporal correlations, which will cause information loss and error accumulation in long-term temporal dependency, and

*Resrach supported by the national key research project Railroad Comprehensive Efficiency and Service Level Improvement Technology with High-Speed Railway Network (Grant No. 2018YFB1201403) in China.

Jianmin Li, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: 19114041 @bjtu.edu.cn).

Xinyue Xu, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (corresponding author to provide phone: (010) – 51688547; email: xxy@bjtu.edu.cn).

Rui Shi, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: 18120877@bjtu.edu.cn).

Xin Ding, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: 20125709@bjtu.edu.cn).



2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564600

cannot accurately capture the dynamic dependence of temporal correlation [17].

Therefore, this paper target is multi-step train delay prediction. A spatial-temporal graph convolutional network (GCN) to sequence model (STG2Seq) is proposed to predict train arrival delay. The main contributions of this paper are summarized as follows: (1) The STG2Seq relies on graph convolution structure, which can capture the spatial-temporal correlations for multi-step prediction between the train arrival delays and various railway system features from large-scale multi-attribute data. (2) An attention-based output module is constructed to splice the output of the long-term and short-term encoder, and then extract important information based on channel attention mechanism. (3) The proposed model is validated using real-world train operation records data from Wuhan-Guangzhou (W-G) HSR, and numerical results show that STG2Seq model is superior to benchmark models.

The remainder of this paper is organized as follows: In Section II, the proposed STG2Seq model for train arrival delay prediction is described in detail. The train delay data and evaluation measurements are described in Section III. In Section IV, the experiment is set up and the prediction performance of the proposed method is analyzed. Finally, the conclusions are presented in Section V.

## II. METHODOLOGY

### A. Problem definition

In this paper, we design a timetable $G$ for HSR line, which includes the scheduled arrival and departure time at each station $S_P$, $S_P \in \{S_1, S_2, ..., S_N\}$, where $S_1$ is the origin station, and $S_N$ is the destination station. Suppose $F$ continuous train runs from $S_1$ to $S_N$, denoted as $1, ..., I, ..., F$, where train $I$ is the interested train [9]. Therefore, the prediction problem of train arrival delay at station-level can be formulated as follows:

**Problem:** Given a timetable $G$ over the previous $P$ stations, the prediction objective of this study is to predict train arrival delay for all trains at the next station, denoted as $y_{I,P+1}$ for one-step forecast, or the train arrival delay for all trains over $\tau$ stations, denoted as $y_{I,P+\tau}$ for multi-step forecast [9].

### B. Overview proposed model

In this section, we describe the proposed STG2Seq method for train delay prediction in detail. The proposed method is divided into three parts, and the specific structure is shown in Figure 1.

(1) Preprocessing of train delay data: first, we fill the incomplete historical data with the mean value of adjacent records. Second, the abnormal observations are replaced by the corresponding modal values, such as negative travel time. Finally, the encoded data is used as the input of GCN layers.

(2) Long-term and short-term encoders: a number of GCN layers are used to form GGCM. Further, long-term encoder and short-term encoder are constructed by stacking GGCM respectively, which reduces the iteration steps of the model in the training process.

(3) Attention- based output module: The output of the long-term and short-term encoder is spliced to get a 3D tensor, and then the key and important information is extracted based on the channel attention mechanism, so that the model can make more accurate judgments and improve the accuracy of prediction. Finally, the loss function is used to evaluate the performance of the prediction model.

The long-term and short-term encoders and attention-based output module are described in detail in the next sub-sections.

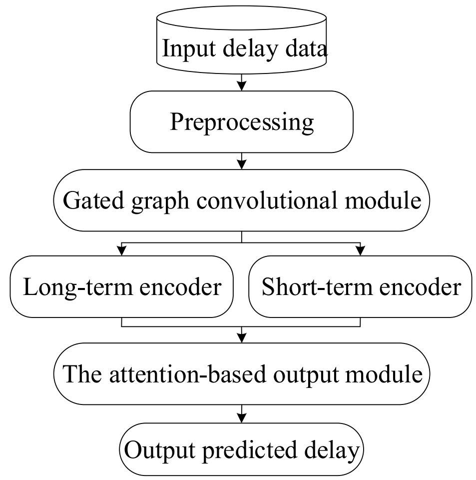


Figure 1. The structure of the STG2Seq model

### C. Long-term and short-term encoders

In this section, we focus on long-term encoder and short-term encoder. The significance of studying the long-term and short-term encoder is to improve the performance of the model in the multiple time steps train delay prediction, and make up for the defects of information oblivion and error accumulation in the RNN based encoder-decoder structure prediction methods.

The input of long-term encoder is the delay data of historical train in $h$ time step, and its purpose is to learn the historical spatial-temporal correlation. Furthermore, the long-term encoder is composed of multiple GGCM. Each GGCM can extract spatial-temporal correlation. Compared with RNN structure model, the iterative time can be reduced and the spatial-temporal correlation can be effectively captured by stacking GGCM [17]. However, the purpose of the short-term encoder is to integrate the predicted train delays for multi-step prediction. The difference between the short-term encoder and the long-term encoder is only in the length of the encoded historical data. The short-term encoder is also composed of multiple GGCM [17].

Secondly, multiple GCN layers are superimposed to form GGCM. In other words, the core of GGCM is GCN. GCN is a method to extract features from graph data, which is an extension of CNN on graph domain (i.e., vertex domain or spectral domain) [18], [19]. Given a graph $G = (V, E)$, the input of GCN is a $N \times F$ feature matrix $X$, where $N$ is the number of nodes, $F$ is the input feature number of each node. Propagation module and output module are the core of GCN. The propagation rules between layers of GCN are as follows [19]:

$$\hat{A}=A+I \tag{1}$$

$$H^{l+1} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{l}W^{l}) \tag{2}$$

where $A$ is a $N \times N$ adjacency matrix of $G$, $I$ is the identity matrix. $H^{l+1}$ and $H^{l}$ represent the feature matrix of layer $l+1$ and layer $l$ respectively, $\sigma$ is the sigmoid function, $\hat{D}$ is the degree matrix of $\hat{A}$, and $W^{l}$ is the weight matrix of $l$ layer. In addition, $A$ is calculated by the following formula:

$$A = \begin{cases} 1, \ if \ Similarity_{di,dj} > \varepsilon \\ 0, \ otherwise \end{cases} \tag{3}$$

where $\varepsilon$ is the threshold. Here, we use Pearson correlation coefficient to quantify the similarity between train delays. The specific framework of GCN is shown in Figure 2.

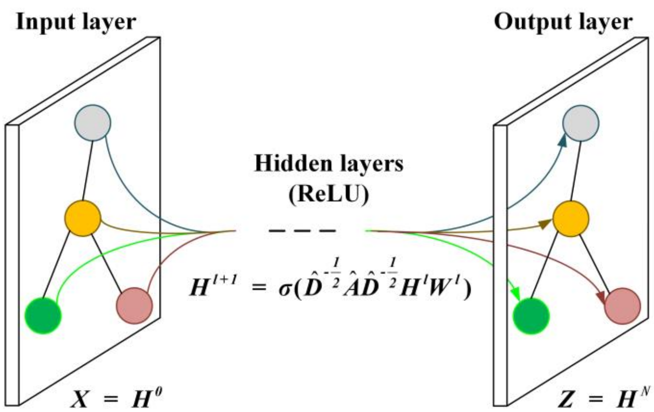


Figure 2. The framework of a graph convolutional network

## D. Attention-based output module

Attention mechanism allows the model to invest more attention resources in local areas to obtain more detailed information of the target and suppress other unimportant information. Attention mechanism has been successfully applied in many fields, such as text feature extraction [11] and facial expression recognition [20]. In this paper, attention mechanism is introduced to enhance the feature extraction ability of channel dimension, and then attention-based output module is formed. The main purpose of the attention-based output module is to splice the output of the long-term and short-term encoder, and then extract important information according to channel attention mechanism. Finally, the optimal predictive value is output. The channel attention is calculated as follows:

$$\beta = softmax(tanh(YW + EW + b)) \tag{4}$$

$$Y_{\beta} = \sum_{i=1}^{O} \beta^{i} y_{i} \tag{5}$$

where $\beta$ indicates the importance score, $W$ and $b$ are transformation matrices, $y_i$ is the demand data of each historical time step, and $Y_{\beta}$ is the predicted train delay demand.

# III. Datasets and Evaluation Measurements

## A. Data description and analysis

In this paper, the real train operation data of W-G HSR is used to evaluate the proposed method. W-G HSR has 17 stations and 16 sections, with a total length of 1,096 kilometers. It is one of the busiest HSR lines in China, passing through big cities such as Wuhan, Changsha and Guangzhou, as illustrated in Figure 2. In this paper, the dataset consists of 1500 train operation records from W-G HSR line from January to March 2019. Each train operation record data includes the train numbers, dates, actual/planned departure/arrival time of each train at each station, as shown in Table I.


Figure 3. Map of the W-G HSR line in china

TABLE I. AN EXAMPLE OF TRAIN OPERATION DATA IN THE DATABASE

| Station | Date | Train | Actual arrival | … |
|---|---|---|---|---|
| YDW | 2019/3/22 | G99 | 20:51 | … |
| LCE | 2019/1/30 | G821 | 18:45 | … |
| HSW | 2019/3/10 | G839 | 19:51 | … |
| YYE | 2019/2/13 | G1526 | 21:28 | … |

## B. Evaluation Measurements

In this paper, three common metrics are used to evaluate the prediction performance of the proposed method for train arrival delay, namely: Root Mean Squared Error (*RMSE*), Mean Absolute Error (*MAE*), and R-squared ($R^2$). Specifically, these indicators are calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (6)$$

$$MAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N} \quad (7)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})} \quad (8)$$

where $y_i$ and $\hat{y}_i$ represent observed and predicted values, respectively.

## IV. Prediction and Results Analysis

### A. Performance analysis

In this section, the implementation details and performance analysis for the proposed STC2Seq method are provided. The performance of STC2Seq model is mainly affected by some hyper-parameters [9]. Here, we refer to other successful studies to set the hyper-parameters of this study, such as the Adam optimizer [21], learning rate [19], activation function [22], training Epochs, and other. All the hyper-parameters used in the STG2seq model are listed in Table II. Furthermore, we implemented all the model on a PC with Intel (R) Core (TM) i7-8565U CPU @ 1.80 GHz 1.99 GHz and 16.0 GB RAM, and python with TensorFlow 1.8 was used.

TABLE II. Structure and Paramters of STG2Seq

| Category | Content |
|---|---|
| GCN model | 2 hidden layers, each with 64 units. |
| Optimizer | Adam |
| Activation function | GLU |
| Learning rate | $1\times10^{-3}$ |
| Training epochs | 300 |
| Mini-batch | 72 |

In this paper, based on the delay data of the first 5 stations, the train arrival delay of the next station and the next $\tau$ station are predicted respectively. Further, we input the training set data into STG2Seq model, and train the proposed prediction model through continuous learning of the training set data. Here, we select 20% of the data set as the test set and 80% as the training set. In the process of training, we define the loss function as the sum of the mean square error of the actual value and the predicted value of train delay for $\tau$ time steps [17], written as:

$$L(W_\theta) = \sum_{t=1}^{t=\tau}\left\|D_T - \hat{D}_T\right\|^2 \quad (9)$$

where $W_\theta$ indicates all the learnable parameters in the neural network.

In the training process, a reasonable number of training epochs will promote the prediction performance of the model. Too few epochs may lead to insufficient model fitting and underfitting, while too many epochs may lead to overfitting and weak generalization ability. Here, after our continuous attempts, we finally set the number of epochs to 300, and found that the model showed the best performance. In addition, we also discuss the fitting effect of the proposed model under different activation functions, and the results show that GLU can promote the model more positively than sigmoid activation function. The model automatically saves the optimal model in the training process, and draws the learning curve as shown in Figure 4. As can be seen from the figure, when the number of iterations is between 0 and 10, the value of loss function decreases rapidly. The number of iterations is between 10 and 250, the value of loss function decreases slowly, and then tends to be stable after 250.

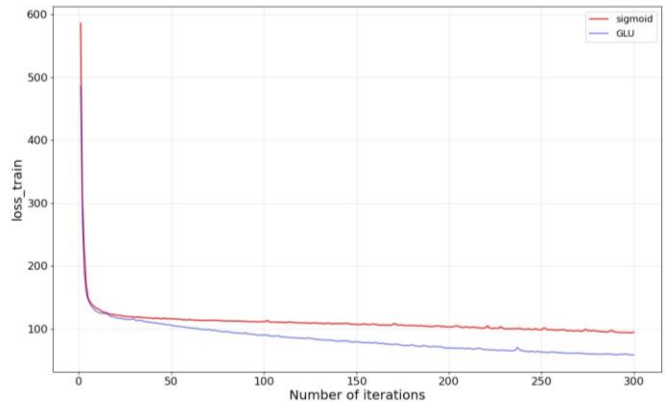


Figure 4. Learning curve under different activation functions

### B. Comparative analysis with other models

In order to compare the performance of our proposed method, the temporal graph convolutional network (T-GCN), k-nearest neighbor (KNN), and gated recurrent unit (GRU) are used as baseline models.

(1) T-GCN is an excellent spatial-temporal graph convolutional network for the problem of prediction. T-GCN mainly uses Gated Recurrent Unit to model temporal correlation [19]. The hyper-parameters are list as follows: learning rate=0.001, hidden units =20, train rate=0.8, batch size=72.

(2) KNN algorithm is a new and highly flexible model method without parameter estimation. Because it is suitable to deal with classification and regression problems, it is widely used in different fields [23]. The hyper-parameters are set as follows: N_neighbors=10, Weights=uniform, Leaf_size=30, P=2.

(3) GRU is a kind of deep learning model, which has the ability to capture temporal correlation, and is very suitable for dealing with the problems highly related to time series [24]. The hyper-parameters are set as follows: learning rate=0.01, hidden units =48, train rate=0.8, batch size=64.

Next, the performances of T-GCN, KNN, GRU and STG2Seq are compared by using *RMSE*, *MAE* and $R^2$. The calculation results of each metric in test set are listed in Table III. The numerical results show that the predicted results are basically consistent in all the tests, and STG2Seq model outperforms T-GCN, KNN, and GRU models (*RMSE*= 4.071, *MAE*=1.754, $R^2$ =0.868). Therefore, the proposed method is suitable for train arrival delay prediction. Furthermore, it can be seen from Table III that the proposed prediction model using GLU activation function does show better performance than sigmoid activation function.

TABLE III. THE PREDICTION ERROR COMPARISON OF EACH MODEL'S TEST SET

| Method | RMSE | MAE | $R^2$ |
|---|---|---|---|
| T-GCN | 4.284 | 1.810 | 0.792 |
| KNN | 4.149 | 2.136 | 0.737 |
| GRU | 4.432 | 1.789 | 0.794 |
| $STG2Seq_{(sigmoid)}$ | 4.350 | 1.883 | 0.856 |
| **$STG2Seq_{(GLU)}$** | **4.071** | **1.754** | **0.868** |

Further, we compare the performance of the proposed method with that of the benchmark methods at the station level. The evaluation metric calculation results of each model at each station are shown in Figure 5-7. Overall, we can observe that the proposed model consistently achieves the best performance and obviously better than other models. More specifically, the proposed method has the minimum *RMSE*, *MAE*, and $R^2$ values for all delayed trains at each station. Furthermore, compared with the T-GCN, KNN, GRU model, our method gains 3.4%, 2.8% and 7.7% relative improvements in *RMSE*; 1.1%, 17.1%, and 6.8% relative improvements in *MAE*; 5.6%, 7.8%, and 7.8% relative improvements in $R^2$, respectively. The results indicate that the proposed method is suitable for solving train delay problems at the station level.

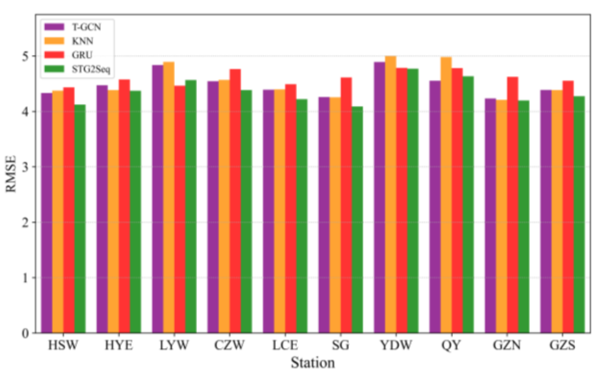


Figure 5. Comparison of the *RMSE* values at different stations

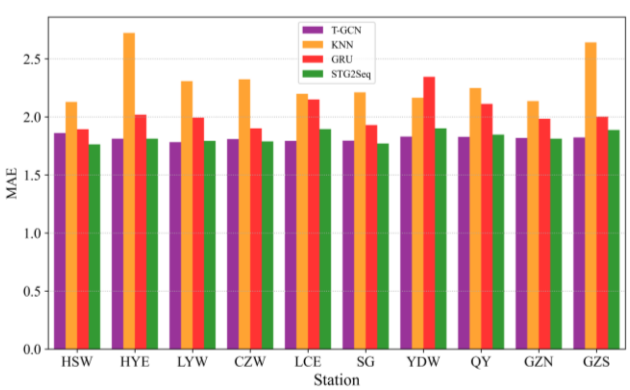


Figure 6. Comparison of the *MAE* values at different stations

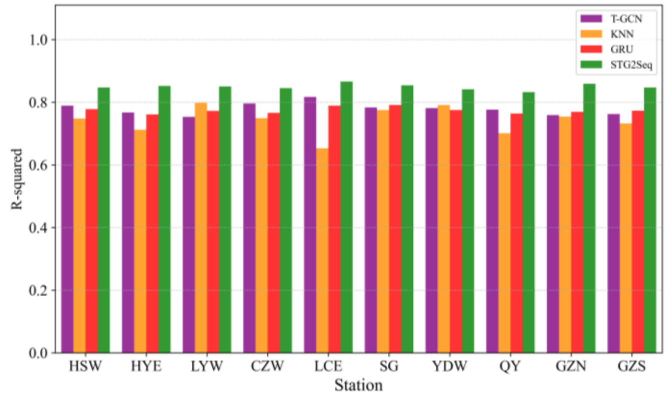


Figure 7. Comparison of the $R^2$ values at different stations

Finally, we compare the proposed model with T-GCN and GRU, which are also capable of conducting multi-step prediction. Each method predicts the train arrival delays in the following 3-time steps. The experimental results about MAE and *RMSE* in W-G HSR datasets are shown in Table IV. The numerical results show that the proposed model also shows good performance for multi-step train delay prediction (*RMSE*: step1=4.071, step2=4.694, step3=5.492; *MAE*: step1=1.754, step2=2.021, step3=2.478), and is superior to T-GCN and GRU. Therefore, the proposed method is suitable for multi-step prediction of train arrival delay.

TABLE IV. EVALUATION OF DIFFERENT FOR MULTI-STEP PREDICTION

| Model | RMSE | | | MAE | | |
|---|---|---|---|---|---|---|
| | step1 | step2 | step3 | step1 | step2 | step3 |
| T-GCN | 4.384 | 5.274 | 5.792 | 1.810 | 2.206 | 2.551 |
| GRU | 4.432 | 5.081 | 5.623 | 1.789 | 2.155 | 2.566 |
| **STG2Seq** | **4.071** | **4.694** | **5.492** | **1.754** | **2.021** | **2.478** |

## V. CONCLUSION

In this paper, a spatial-temporal graph convolutional network to sequence model (STG2Seq) was proposed to predict multi-step train arrival delays. For one thing, the introduction of long-term and short-term encoders enables the multi-step prediction of train delay, and further analyzes the relationship between train arrival delays and various railway system characteristics. For another, the proposed method uses attention mechanism to consider the dynamic property of temporal correlation. In addition, the predictive performance of proposed method for train delay prediction is evaluated based on real-world HSR lines, and T-GCN, KNN, and GRU were used as the baseline model. The main findings can be summarized as follows: (1) The STG2Seq can suitably fit the delay data of high-speed trains at different spatial and temporal distributions, and has a great capacity for solving the train delay prediction problem. (2) The proposed method is associated with smaller forecasting errors for the real-world HSR lines in China (*RMAE* = 4.071, *MAE* = 1.754, and $R^2$ = 0.868). It is verified that the proposed method has excellent prediction accuracy. (3) Based on the common indicators, the proposed model outperforms other state-of-the-art models.

## REFERENCES


[1] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transp. Res. Part C Emerg. Technol.*, vol. 56, pp. 251–262, 2015, doi: 10.1016/j.trc.2015.04.004.

[2] C. Wen, W. Mou, P. Huang, and Z. Li, "A Predictive Model of Train Delays on a Railway Line," *J. Forecast.*, vol. 39, no. 3, pp. 470–488, 2020.

[3] P. Huang, C. Wen, L. Fu, Q. Peng, and Y. Tang, "A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems," *Inf. Sci. (Ny).*, vol. 516, pp. 234–253, 2020, doi: 10.1016/j.ins.2019.12.053.

[4] L. Oneto *et al.*, "Train Delay Prediction Systems: A Big Data Analytics Perspective," *Big Data Res.*, vol. 11, pp. 54–64, 2018, doi: 10.1016/j.bdr.2017.05.002.

[5] F. Corman and P. Kecman, "Stochastic prediction of train delays in real-time using Bayesian," *Transp. Res. Part C*, vol. 95, no. October 2017, pp. 599–615, 2018, doi: 10.1016/j.trc.2018.08.003.

[6] M. F. Gorman, "Statistical estimation of railroad congestion delay," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 45, no. 3, pp. 446–456, 2009, doi: 10.1016/j.tre.2008.08.004.

[7] J. Yuan and I. A. Hansen, "Optimizing capacity utilization of stations by estimating knock-on train delays," *Transp. Res. Part B Methodol.*, vol. 41, no. 2, pp. 202–217, 2007, doi: 10.1016/j.trb.2006.02.004.

[8] A. Higgins and E. Kozan, "Modeling Train Delays in Urban Networks," *Transp. Sci.*, vol. 32, no. 4, 1998, doi: 10.1016/j.jrtpm.2011.06.001.

[9] P. Huang *et al.*, "Modeling train operation as sequences: A study of delay prediction with operation and weather data," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 141, 2020, doi: 10.1016/j.tre.2020.102022.

[10] J. Hu and N. Bernd, "Application of Artificial Neuron Network in Analysis of Railway Delays," 2016.

[11] Y. Masoud, M. M. Khoshraftar, and S. Masoud, "Railway passenger train delay prediction via neural network model," *J. Adv. Transp.*, vol. 47, no. 3, pp. 355–368, 2013, doi: 10.1016/j.eswa.2020.113673.

[12] R. Shi and X. Xu, "A Train Arrival Delay Prediction model using XGBoost and Bayesian Optimization*," *2020 IEEE 23rd Int. Conf. Intell. Transp. Syst. ITSC 2020*, 2020, doi: 10.1109/ITSC45102.2020.9294365.

[13] J. Lessan, L. Fu, and C. Wen, "A hybrid Bayesian network model for predicting delays in train operations," *Comput. Ind. Eng.*, vol. 127, no. March 2018, pp. 1214–1222, 2019, doi: 10.1016/j.cie.2018.03.017.

[14] F. J. Martinez-de-Pison, R. Gonzalez-Sendino, A. Aldama, J. Ferreiro-Cabello, and E. Fraile-Garcia, "Hybrid methodology based on Bayesian optimization and GA-PARSIMONY to search for parsimony models by combining hyperparameter optimization and feature selection," *Neurocomputing*, vol. 354, pp. 20–26, 2019, doi: 10.1016/j.neucom.2018.05.136.

[15] Y. Li, X. Xu, J. Li, and R. Shi, "A delay prediction model for high-speed railway: An extreme learning machine tuned via particle swarm optimization," *2020 IEEE 23rd Int. Conf. Intell. Transp. Syst. ITSC 2020*, 2020, doi: 10.1109/ITSC45102.2020.9294457.

[16] L. Oneto *et al.*, "Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 10, pp. 2754–2767, 2017, doi: 10.1109/TSMC.2017.2693209.

[17] L. Bai, Y. Lina, K. Salil S., W. Xianzhi, and S. Quan Z., "STG2Seq: Spatial-temporal Graph to Sequence Model for Multi-step Passenger Demand Forecasting."

[18] L. Qiangsheng, X. Xinyue, and L. Liqiang, "Data-Driven Short-Term Passenger Flow Prediction Model for Urban Rail Transit," vol. 41, pp. 153–162, 2020.

[19] J. Tang, J. Liang, F. Liu, J. Hao, and Y. Wang, "Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network," *Transp. Res. Part C Emerg. Technol.*, vol. 124, no. January, p. 102951, 2021, doi: 10.1016/j.trc.2020.102951.

[20] J. Li, K. Jin, D. Zhou, N. Kubota, and Z. Ju, "Attention mechanism-based CNN for facial expression recognition," *Neurocomputing*, vol. 411, pp. 340–350, 2020, doi: 10.1016/j.neucom.2020.06.014.

[21] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An improved bayesian combination model for short-term traffic prediction with deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1332–1342, 2020, doi: 10.1109/TITS.2019.2939290.

[22] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *34th Int. Conf. Mach. Learn. ICML 2017*, vol. 2, pp. 1551–1559, 2017.

[23] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.

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