Railway Delay Prediction with Spatial-Temporal Graph Convolutional Networks

Jacob S.W. Heglund, Panukorn Taleongpong, Simon Hu *Member, IEEE*, Huy T. Tran *Member, IEEE*

***Abstract*—Cascading delays that propagate from a primary source along a railway network are an immediate concern for British railway systems. Complex nonlinear interactions between various spatio-temporal variables govern the propagation of these delays which can quickly spread throughout railway networks, causing further severe disruptions. To better understand the effects of these nonlinear interactions, we present a novel, graph-based formulation of a subset of the British railway network. Using this graph-based formulation, we apply the Spatial-Temporal Graph Convolutional Network (STGCN) model to predict cascading delays throughout the railway network. We find that this model outperforms other statistical models which do not explicitly account for interactions on the rail network, thus showing the value of a Graph Neural Network (GNN) approach in predicting delays for the British railway system.**

***Index Terms*—Intelligent Transport Systems, Railway, Machine Learning, Deep Learning, Big Data**

I. Introduction

THE British railway, dating back to 1825, is the oldest operational railway network in the world, the majority of which is owned and maintained by Network Rail. Train operating companies (TOCs) provide transportation services to be run on the Network Rail's infrastructure and are members of the National Rail, whose purpose is to promote these services and provide the public with information on all passenger rail services [1]. The British rail industry is currently experiencing a stagnation in performance affecting a rapidly growing commuter population on a daily basis. The Rail Research UK Association [2] predicts the increase of cascading delays, train delays that are a result of its prior delays or the propagation of delay from any other train, from 600,000 minutes annually to 800,000 minutes annually in the last five years. With the number of passengers travelling on British train networks almost doubling from 1 billion to 1.7 billion in the past two decades [3], this trend will only continuously worsen unless appropriate measures are put into place. Due to the complex nonlinear spatio-temporal variable interactions governing the propagation of these delays, they are inherently difficult to predict.

The work is supported in part by the Institute for Sustainability, Energy, and Environment (iSEE) at the University of Illinois at Urbana-Champaign and the Zhejiang University/University of Illinois at Urbana-Champaign Institute. It was led by Principal Supervisors Simon Hu and Huy T. Tran.

Jacob S.W. Heglund is with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (email: jheglun2@illinois.edu).

Panukorn Taleongpong is with the Center for Transport Studies, Department of Civil and Environmental Engineering, Imperial College London, SW7 2BU, UK (email: Panukorn_T@live.com).

Hu, S is the Corresponding Author with the School of Civil Engineering, ZJU-UIUC Institute, Zhejiang University, Zhejiang,314400, China (phone: +868867516624; e-mail: simonhu@zju.edu.cn).

Huy T. Tran is with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (email: huytran1@illinois.edu).

Many mathematical and statistical models have been established in the literature to predict delays and understand its mechanism of propagation throughout a railway network. [4] proposed utilization of activity graphs for the computation of delay propagation and used distribution functions to describe the random nature of delays. [5] proposed the prediction of delay propagation along a transportation network with stochasticity induced in resource conflict as opposed to waiting for connecting trains. Trains in the network are assigned static priority classes that shall be used to resolve resource conflict. [6] proposed the modelling of a railway system as a linear system in max-plus algebra with zero-order dynamics that represent delay propagation. [7] proposed different distributions for eleven delay types ranging from bad weather to fault in tracks and utilized maximum likelihood estimation (MLE) and the Kolmogorov-Smirnov test (K-S) to evaluate each proposed distribution. [8] developed an algorithm that analyzes real-world data to identify delays and cascading delays defined according to various conditions, returning a network of dynamic delay propagation. [9] utilized the closed episode algorithm to mine cascading delays through a Belgian railway network focusing on specific reference points throughout the network. Finally, [10] developed three regression models to predict the delay of trains at stations, each of which introduced different assumptions about the current delay and previous delays.

With recent advancements in the field of Machine Learning (ML), many have explored the use of ML models to predict delays and understand the mechanism of delay propagation in the aviation industry [11]; however, only a handful of projects utilize ML models in the same way in the railway industry. [12] proposed several regression models including random forests and feed-forward neural networks toward the problem of estimating time of arrival in United States rail networks. [13] proposed the adoption of recurrent neural networks

(RNNs) alongside Irish Rail System data with labeled delay types to perform a one station step delay forecast. [14] produced a train delay prediction system, forecasting the time taken for the train to reach its next checkpoint considering its scheduled journey up to terminal station. [15] utilized weather records, historical delays, and train schedules to identify delay-inducing factors, and utilized gradient-boosted regression trees to predict delays along the Beijing Guanzhou line. Finally, [16] explored weather data for delay prediction in rail networks through the use of kernel-based methods, extreme learning machines, and ensemble methods.

While these efforts demonstrate the value of ML for predicting delays in railway systems, previous approaches have typically focused on small or single-line railways, and do not explicitly consider the connections between elements in the rail network, limiting their use in more complex railway networks. Graphical Neural Networks (GNNs) offer an approach for handling such spatial complexities by explicitly modeling network dependencies and leveraging them for prediction. Two specific GNN architectures, the Spatial-Temporal Graph Convolutional Network (STGCN) [17] and its attention-based extension [18], have particularly shown promise for the prediction of delays in transportation systems. [17] and [18] showed that GNN models are able to provide high-resolution predictions of delays in traffic flow by capturing spatial patterns with graph-based convolutions and temporal patterns with temporal convolutions.

This paper applies the STGCN model to predict delays within a portion of the British rail network where Didcot Parkway and Long Paddington serve as gateway stations. We formulate the rail network as a line graph, where nodes represent rail links and edges represent connecting stations. We then apply the STGCN model to this formulation and compare its performance to alternative statistical models. We find that the STGCN model outperforms those alternatives in mean absolute error (MAE) and RMSE for forecast intervals ranging from ten to 60 minutes. Our results demonstrate the value of explicitly leveraging the rail network structure within prediction models and identify several promising directions for using GNNs to predict rail delays.

II. Preliminaries

A. Delay Prediction on Railway Graphs

We formulate the prediction of delays on the rail network as a time series regression problem in which observed delays on links, or connections between stations, at the previous N_{past} time steps are used to predict the most likely delay at the $t + N_{\text{future}}$ time step. We use the following definition for links of the rail network:

Definition. Rail Link: A rail link AB exists between Station A and Station B if a train on the rail network does not pass through any other station on the network in between Station A and Station B.

Based on this definition, we formally state the regression problem on the rail network as,

$$\hat{y}_{t+N_{\text{future}}} = \underset{v_{t+N_{\text{future}}}}{\operatorname{argmax}} \log P(v_{t+N_{\text{future}}} | v_{t-N_{\text{past}}}, ..., v_t) \tag{1}$$

where $v_t \in \mathbb{R}^{N \times F}$ is a tensor of F delay features on N links of the rail network at time t, and $\hat{y}_t \in \mathbb{R}^{N \times F}$ is a tensor of model predictions at time t. Note that we only consider one feature, link delay, for this study (i.e., $F = 1$).

We then represent the rail network as an undirected and attributed graph $G(V, E, X(t))$, defined by nodes V, edges E, and time-varying node features $X \in \mathbb{R}^{N \times F \times T}$, where $N = |V|$ is the number of nodes in the graph, and T is the total number of time steps in the dataset. This graph may be represented by its adjacency matrix A_G, defined as,

$$\begin{aligned} A_G(i,j) &= 1 \quad \text{if stations } i \text{ and } j \text{ share a link,} \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{2}$$

Note that while the node features are time-dependent, the underlying graph structure remains static throughout the data period.

This graph formulation considers the delays of links in the rail network as edge-wise features of G. While recent work has explored the use of edge-wise features for graph prediction, these architectures often do so in an effort to simultaneously leverage node- and edge-wise features. Since we only consider one set of features (i.e., link delays) we do not require such an architecture. We therefore invert the nodes and edges of G to produce a line graph of the rail network $\mathcal{L}$ to enable the use of architectures with only node-wise features. This line graph then has an adjacency matrix $A_{\mathcal{L}}$, defined as,

$$\begin{aligned} A_{\mathcal{L}}(i,j) &= 1 \quad \text{if links } i \text{ and } j \text{ are connected by a station,} \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{3}$$

We use this line graph to capture spatial relationships within the data in our proposed model architectures.

B. Graphical Neural Networks

ML methods have shown promise for predicting delays in transportation systems. These methods typically leverage convolution operations to capture spatial relationships within the data. GNNs extend these methods to be applicable for graph-structured data by specifically leveraging graph convolutions to propagate information between neighboring nodes and embed provided graph features into a latent space. This embedding provides a high-level representation of the data, which is then typically combined with a multi-layer perceptron or softmax output layer to provide node-level

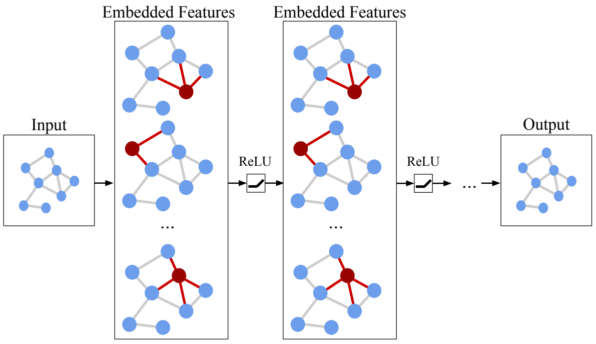

Fig. 1. A generic GNN architecture implementing repeated layers of graph convolutions and ReLU activations. Red is used to denote information propagation between nodes. The graph convolution creates a set of embedded node-wise features used for prediction.

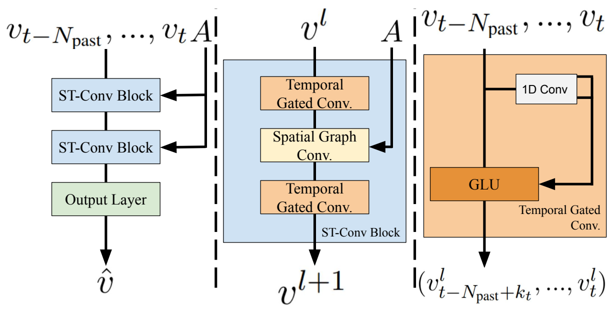

Fig. 2. The STGCN model as developed in [17]. The overall model architecture is shown on the left, the ST-Conv block in the middle, and the temporal gated conv block on the right.

predictions. A generic GNN model is shown in Figure 1. See [19] and [20] for surveys of GNNs and their applications.

III. Proposed Model

A. Network Architecture

Based on the line graph formulation, we select an architecture that leverages node-wise features for the graph prediction problem. We use the STGCN model for this effort because it explicitly considers spatial and temporal dimensions of the rail network data [17]. The model architecture is summarized in Figure 2. The architecture contains two stacked spatio-temporal convolutional blocks (ST-Conv blocks) followed by an output block, which is itself composed of a temporal convolution followed by a fully-connected layer. We use the L2 loss function to train this architecture, defined as,

$$L(\hat{y};\theta) = \sum_t ||f(v_{t-N_{\text{past}}},\dots,v_t;\theta) - v_{t+N_{\text{future}}}||^2 \quad (4)$$

where θ are trainable model parameters, $v_{t+N_{\text{future}}}$ is the ground truth, and $f(\cdot)$ denotes the model's prediction. The following sub-sections provide details of the model at the level of the individual spatial and temporal convolutional blocks.

B. Convolution in the Spatial Dimension

The ST-Conv blocks of the STGCN architecture leverage graph convolutions to capture spatial relationships in the data. Spectral graph theory provides one method (i.e., the graph Fourier transform) for generalizing the convolution operation for graph-structured data. The analysis focuses on the eigenvalues of the normalized graph Laplacian matrix, given as,

$$L = I_N - D^{-1/2}AD^{-1/2} \quad (5)$$

where $I_N \in \mathbb{R}^{(N\times N)}$ is the N-dimensional identity matrix which adds self-loop connectivity to the adjacency, $A \in \mathbb{R}^{(N\times N)}$ is the graph adjacency matrix, and $D \in \mathbb{R}^{(N\times N)}$ is the diagonal degree matrix of A such that $D_{ii} := \sum_j A_{ij}$.

The graph convolution "$*_G$" is defined as the multiplication of the graph signal $x \in \mathbb{R}^N$ with kernel Θ, such that,

$$\Theta *_G x = \Theta(L)x = \Theta(U\Lambda U^T)x = U\Theta(\Lambda)U^Tx \quad (6)$$

where the graph Fourier basis $U \in \mathbb{R}^{(N\times N)}$ is the matrix of eigenvectors of the normalized graph Laplacian, $\Lambda \in \mathbb{R}^{(N\times N)}$ is the diagonal matrix of eigenvalues of L, and kernel $\Theta(\Lambda)$ is a diagonal matrix. Note that we denote the convolution operation on any generic graph G, which in our implementation is a rail network line graph $\mathcal{L}$.

Computation of Θ requires $\mathcal{O}(n^2)$ operations, making it computationally inefficient for large-scale graphs. However [21] introduces an approximation that restricts the graph kernel Θ to the set of Chebyshev Polynomials, and [22] introduces as a first-order approximation for the graph kernel. Both of these approximations are utilized in the STGCN architecture, after being generalized for use with multi-dimensional tensors. For brevity we do not include the details of the approximations or the generalization of graph convolution in this paper; however, more details may be found in [23].

C. Convolution in the Temporal Dimension

The ST-Conv blocks also leverage a convolution to capture temporal relationships in the data. Recurrent Neural Networks (RNNs) are often used for this purpose; however, these networks can be difficult to train due to the "vanishing gradient" problem. Additionally, recent papers [24] have shown that a 1D convolution along the temporal dimension of data can be more effective than an RNN on shorter sequences, while at the same time being quicker to train. As shown in Figure 2 (right), the temporal convolutional layer of each ST-Conv block contains a 1D causal convolution with kernel of size k_t and a gated-linear unit (GLU) nonlinear activation. Similar to the gating present in RNN models, namely LSTM and GRU, the nonlinear activation provides a gating which determines importance of past inputs on future predictions. The resulting temporal convolution is defined as,

TABLE I
SERVICE METRICS PROVIDED BY DARWIN'S HSP

Key	Description
Origin Location	Computer Reservation System (CRS) code of origin
Destination Location	CRS code of destination
gbtt ptd	Public departure time at departure station
gbtt pta	Public arrival time at destination station
TOC code	Code of train operating company
RIDs	Train ID
Matched services	List of all train RIDs
Tolerance Value	Tolerance for difference between actual and public arrival time
Num not tolerance	Number of trains outside the tolerance
Num tolerance	Number of trains within the tolerance

TABLE II
SERVICE DETAILS PROVIDED BY DARWIN'S HSP

Key	Description
Date of service	Date of service of the specified train RID
TOC Code	Code of train operating company
RID	Inputted RID
Location	CRS code of train location
gbtt ptd	Public departure time
gbtt pta	Public arrival time
Actual td	Actual departure time
Actual ta	Actual arrival time
Late canc reason	Code that specifies late or cancellation reason

$$\Gamma *_T Y = P \odot \sigma(Q) \tag{7}$$

where P and Q result from splitting the input of the temporal block along the "channels" dimension. Further details of the temporal convolution, including generalization to 3D tensors, are provided in [17].

D. Spatio-Temporal Convolutional Block

The ST-Conv blocks are constructed by combining these graph and temporal convolutions to capture spatio-temporal behaviors. The lth ST-Conv block is then given as,

$$v^{l+1} = \Gamma_1^l *_T \mathrm{ReLU}(\Theta^l *_G (\Gamma_0^l *_T v^l)) \tag{8}$$

where Γ_0^l and Γ_1^l are the temporal kernels within block l, Θ^l is the spectral kernel of the graph convolution, and ReLU denotes a rectified linear unit activation.

IV. EXPERIMENT DESCRIPTION

A. Data Source

The dataset used in this paper is provided through Darwin, Great Britain's official railway information engine [25]. Specifically, the application programming interface (API) we utilize is the Historical Service Performance (HSP) API [26]. This API provides two datasets through two separate calls in Javascript Object Notation (JSON) Format, which are used in conjunction with each other. The first call, Service Metrics, requires the origin and destination stations, first departure and final arrival times, and start and end dates to be defined as inputs. The second call, Service Details, requires train IDs provided by the Service Metrics API as input. The data received through the Service Metrics call is outlined in Table I and data received through the Service Details call is outlined in Table II.

All time values provided by the Service Details API are accurate to the nearest minute and include all origin-destination trips that pass through the gateway origin and destination stations. For this paper, we select Didcot Parkway and London Paddington as the gateway stations; i.e., all train journeys that include both these stations in their schedule in both inbound and outbound directions are included in the dataset. Journeys between these stations were chosen due to their notoriety in providing prevalent delayed services [27]. Figure 3 shows the rail network stations included for the gateway stations of Didcot Parkway and London Paddington. In the inbound direction, Darwin provides 10,767 journeys in 2016 and 10,742 journeys in 2017 initiating at various stations, passing through Didcot Parkway, and terminating at London Paddington. In the outbound direction, Darwin provides 9,069 journeys in 2016 and 8,969 journeys in 2017 initiating at London Paddington and passing through Didcot Parkway on the way to their respective destination stations.

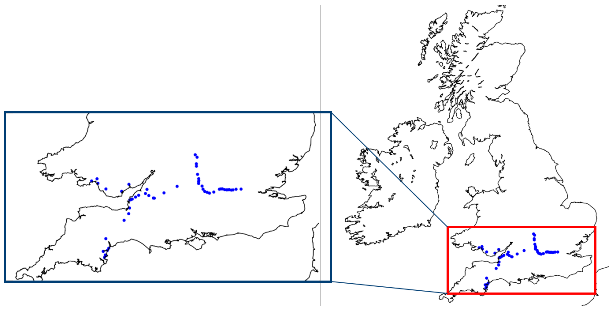

Fig. 3. The subset of the British rail network with Didcot Parkway and London Paddington as the gateway stations.

B. Data Preprocessing

Given the raw data provided by Darwin, we include train journeys on non-holiday weekdays starting between 5:30 AM and 12:00 PM from 2016 and 2017. This time range was selected to capture the mechanics of delay propagation during peak usage of the rail network. We construct a line graph of the rail network $\mathcal{L}$ by setting links between stations as nodes of the graph and stations connecting links as edges. For consistency of the graph structure, we remove any links that are included in one year but not the other. We also only consider inbound trips for this set of experiments; i.e., we only include the trips beginning at some station, passing through Didcot Parkway, and terminating at London

Paddington. Finally, we remove stations that serve an average of fewer than one train per day in order to reduce noise in the graph signal. We use the NetworkX library to calculate the line graph from G. The rail links included in this graph, and their usage during the considered time period.

Our model uses the arrival delay of trains passing through links on the rail network as its feature. This feature is used in order to measure the congestion experienced on each link of the rail network. Arrival delay is defined as $d_{\text{arr}} = t_{\text{arr, sched}} - t_{\text{arr, actual}}$, where $t_{\text{arr, sched}}$ and $t_{\text{arr, actual}}$ are the scheduled and actual arrival time of the train, respectively. We attribute the experienced arrival delay to each link traversed by the train in between its origin and destination stations. The following definitions explain the delay attribution process:

Definition. Route: A route is the set of rail links traversed by a train in between stations at which it stops. We denote the number of links in a route as n_{links}. Note that a train may traverse a link as part of a route without stopping at either of the terminating stations on that link.

For example, consider rail network A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D and a train which departs from A, does not stop at B or C, and stops at D. The train's route for this section of its trip would be (AB, BC, CD).

Definition. Links Traversed during Time Period (t_0, t_1): Consider rail network A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D and a train which departs from A, does not stop at B or C, and stops at D. Links AB, BC, and CD along route (AB, BC, CD) are considered traversed during (t_0, t_1) if any of the following are true:

i) the time at which the train departed from Station A falls within (t_0, t_1),

ii) the time at which the train arrives at Station D falls within (t_0, t_1),

iii) the average time at which the train was traversing the route falls within (t_0, t_1).

Definition. Link Attributed Arrival Delay: Denote Link Attributed Arrival Delay as $d_L := \frac{d_{\text{arr}}}{n_{\text{links}}}$. d_L is a feature of the link AB during time period (t_0, t_1) if and only if the link is part of a route that was traversed during (t_0, t_1).

We consider a sequence of node features $(v_{t-N_{\text{past}}}, ..., v_t)$ as our model input, and a single interval $v_{t+N_{\text{future}}}$ as the model output. Since the majority of the routes traversed in the dataset last fewer than 15 minutes, we choose a sampling time interval of 10 minutes. That is, we sample the delay along each rail link at consecutive 10 minute intervals (e.g., [0900, 0910], [0910, 0920], ...). For numerical stability, all features are normalized and the z-score of link attributed delay is used to train the model. Finally, we implement a uniformly sampled 70 / 20 / 10% split of the data for the training, validation, and testing datasets, respectively. All metrics presented in this paper were calculated on the test data which is not observed during model training.

C. Model Settings

For the STGCN model, we use a spatial kernel of size $k_s = 5$ and a temporal kernels of size $k_t = 3$. We use the Chebyshev polynomial approximation of the graph Laplacian, and the channels within each ST-Conv block take a bottleneck form such that the number of channels are given as Block 1 = (1, 32, 64), Block 2 = (64, 32, 128), and Output Block = (128, 1). We train each model for 25 epochs with a batch size of 100 using the ADAM optimizer and L2 loss with an initial learning rate $R = 0.001$. Finally, we implement a learning rate decay of $R \leftarrow 0.1R$ every 10 epochs.

We also implement a multi-layer perceptron (MLP) model for comparison. We use a 3-layer fully-connected model with a 1-node input, 100-node hidden layer, and 1-node output. We use the RELU activation for the first two layers, and a sigmoid activation on the model output. This model is trained for 25 epochs using the same optimizer and loss function as STGCN.

V. Results and Evaluation

We compare the STGCN model's performance to two common statistical methods: linear regression (LR) and MLP. Neither LR nor MLP explicitly model the connections of graph-structured data, so for each node in the graph we optimized a new model for delay prediction. Furthermore, neither LR nor MLP are designed for time series prediction, so the features of each input time step are appended to form a feature vector of size $((N_{\text{past}} * F) \times 1)$. We test each model under multiple $(N_{\text{past}}, N_{\text{future}})$ time step conditions to understand the flexibility of the STGCN model its sensitivity to the input sequence length. We use MAE and RMSE to evaluate our models. These metrics are calculated by first averaging over the nodes of the graph, then averaging over the number of sequences in the dataset, and finally averaging over a set of 5 replicates per model. The results of our experiments are presented in Table III.

We find that STGCN outperforms the other considered models on all test conditions. This result is likely due to the model's ability to capture dependencies between neighboring nodes in the graph via the graph convolution operation, as well as its ability to capture temporal dependencies via the temporal convolution operation. The combination of these operations implicitly models nonlinear cascading delays in the rail network. Our results show that this deep learning architecture can be readily applied to leverage available rail network data and provides more accurate predictions than classical statistical methods.

While our results were calculated on a subset of the British rail network, the STGCN model can easily scale to larger graphs while still capturing local dependencies between nodes. This scalability is due to the use of graph convolutions, which allow the model to output predictions for

TABLE III
ACCURACY METRICS ON RAIL NETWORK DATA

Model		
$N_{past} = 6$	MAE (10 / 30 / 60 min)	RMSE (10 / 30 / 60 min)
LR	0.304 / 0.365 / 0.36	0.69 / 0.834 / 0.847
MLP	0.341 / 0.362 / 0.364	0.966 / 0.915 / 1.096
STGCN	**0.256 / 0.311 / 0.302**	**0.625 / 0.803 / 0.755**
$N_{past} = 12$	MAE (10 / 30 / 60 min)	RMSE (10 / 30 / 60 min)
LR	0.279 / 0.337 / 0.338	0.59 / 0.753 / 0.785
MLP	0.331 / 0.34 / 0.327	0.982 / 0.896 / 0.931
STGCN	**0.25 / 0.282 / 0.27**	**0.539 / 0.713 / 0.669**

every node simultaneously. In comparison, classical statistical models either require a model to be trained for every node or implicitly assume full information propagation amongst nodes in a multiple response formulation. We also found that, while training the STGCN model took longer than the other methods, the STGCN model still trains relatively quickly, requiring on average 20 seconds per epoch for this dataset using a computer with an AMD Ryzen Threadripper 2920X CPU and an NVIDIA RTX 2080 GPU.

VI. CONCLUSION

We present a novel, graph-based formulation of the British rail network. Using this formulation, we apply the STGCN architecture to implicitly model nonlinear cascading delays on the rail network and predict expected delays that trains would experience traversing each link of the rail network. Experiments on real-world rail data show that this architecture provides more accurate predictions than classical statistical models due to its ability to capture both spatial and temporal dimensions of the data.

Future extensions of this work include more thorough comparisons of GNNs with existing models in the railway literature and alternative problem formulations to predict delays on specific routes and more explicitly consider inbound and outbound traffic on the rail network. Further study of the causes and propagation of delay in the rail network should also be included and develop of our models for real-world deployment.

REFERENCES

[1] National Rail, "About national rail enquiries," 7 Mar 2019. [Online]. Available: http://www.nationalrail.co.uk/46383.aspx

[2] RRUKA, "Call for research, data sandbox: Improving network performance," Tech. Rep., Oct 31, 2019.

[3] Office Of Rail and Road, "Passenger and freight rail performance 2018-19 q4 statistical release," Tech. Rep., May 24 2019.

[4] T. Buker and B. Seybold, "Stochastic modelling of delay propagation in large networks," *Journal of Rail Transport Planning and Management*, vol. 2, no. 1, pp. 34–50, 2012.

[5] H. Khadilkar, "Modelling the impact of control strategy on stochastic delay propagation in transportation networks," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 2471–2476.

[6] R. M. P. Goverde, "A delay propagation algorithm for large-scale railway traffic networks," *Transportation Research Part C: Emerging Technologies; 11th IFAC Symposium: The Role of Control*, vol. 18, no. 3, pp. 269–287, 2010.

[7] Y. Yang, P. Huang, Q. Peng, J. Li, and C. Wen, "Statistical delay distribution analysis on high-speed railway trains," *Journal of Modern Transportation*, vol. 27, no. 3, pp. 188–197, Sep 2019.

[8] A. O. Sorensen, A. D. Landmark, N. O. E. Olsson, and A. A. Seim, "Method of analysis for delay propagation in a single-track network," *Journal of Rail Transport Planning and Management*, vol. 7, no. 1, pp. 77–97, 2017.

[9] B. Cule, B. Goethals, S. Tassenoy, and S. Verboven, "Mining train delays," J. Gama, E. Bradley, and J. Hollmn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 113–124.

[10] R. Wang and D. B. Work, "Data driven approaches for passenger train delay estimation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 535–540.

[11] K. Gopalakrishnan and H. Balakrishnan, "A Comparative Analysis of Models for Predicting Delays in Air Traffic Networks," in *Twelfth USA/Europe Air Traffic Manag. Res. Dev. Semin.*, 2017.

[12] W. Barbour, C. Samal, S. Kuppa, A. Dubey, and D. B. Work, "On the data-driven prediction of arrival times for freight trains on us railroads," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2289–2296.

[13] E. Bosscha, "Big data in railway operations: Using artificial neural networks to predict train delay propagation," Ph.D. dissertation, Jun 2016.

[14] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Train delay prediction systems: A big data analytics perspective," *Big Data Research; Selected papers from the 2nd INNS Conference on Big Data: Big Data and Neural Networks*, vol. 11, pp. 54–64, 2018.

[15] P. Wang and Q.-p. Zhang, "Train delay analysis and prediction based on big data fusion," *Transportation Safety and Environment*, vol. 1, no. 1, pp. 79–88, 02 2019.

[16] L. Oneto, E. Fumeo, G. Clerico, R. Canepa, F. Papa, C. Dambra, N. Mazzino, and D. Anguita, "Advanced analytics for train delay prediction systems by including exogenous weather data," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 458–467.

[17] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[18] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, p. 922929, 2019.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[20] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[21] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129 – 150, 2011.

[22] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, ser. ICLR '17, 2017.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[24] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint*, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[25] L. Bleakley, A. Akinola, and R. Fullard, "Rdg information feeds developer pack," Oct 2019. [Online]. Available: https://www.nationalrail.co.uk/46391.aspx

[26] "Darwin data feeds," 2019. [Online]. Available: https://www.nationalrail.co.uk/100296.aspx

[27] Department of Transport, "England and wales top 10 overcrowded train services: Spring and autumn 2015," Tech. Rep., Jul 28 2016.