12-2021

Context-aware graph convolutional network for dynamic origin-destination prediction

Juan NATHANIEL
Singapore Management University, jnathaniel@smu.edu.sg

Baihua ZHENG
Singapore Management University, bhzheng@smu.edu.sg

Context-Aware Graph Convolutional Network for Dynamic Origin-Destination Prediction

Juan Nathaniel
School of Engineering and Applied Science
Columbia University
jn2808@columbia.edu

Baihua Zheng
School of Computing and Information Systems
Singapore Management University
bhzheng@smu.edu.sg

***Abstract*—A robust Origin-Destination (OD) prediction is key to urban mobility. Such a model can minimize the operational risks and improve service availability, among many other upsides. Here, we examine the use of Graph Convolutional Network (GCN) and its hybrid Markov-Chain (GCN-MC) variant to perform a context-aware OD prediction based on a large-scale public transportation dataset collected in Singapore. Compared with the baseline Markov-Chain algorithm and GCN, the proposed hybrid GCN-MC model improves the prediction accuracy by 37% and 12% respectively. Lastly, the addition of temporal and contextual information further improves the performance of the proposed hybrid model by $4-12\%$.**

***Index Terms*—Graph Convolutional Network (GCN); Markov Chain; public transportation; OD prediction; explainable AI (XAI)**

I. Introduction

Mass public transportation is one of the most important infrastructures in major cities, allowing millions of people to move seamlessly. For land-scarce cities, such as Tokyo and Singapore, it is critical to improve transport infrastructure by estimating the future demand for an efficient, reliable, and affordable mobility. OD prediction, a key component of this demand analysis, has received a considerable amount of attention due to a surge in the amount of available public trips data [1, 2, 3, 4, 5]. However, the task is uniquely challenging because the underlying dataset tends to have high dimensionality, exhibits spatial and temporal dependencies, and the prediction itself is sensitive to contexts (eg., time of the day, traffic condition, disruptions) [6]. Therefore, we propose a modeling framework that solves the high-dimensionality, spatiotemporally-dependent OD problem through a Graph Convolutional Network (GCN) and incorporates contexts using a Markov Chain (MC) algorithm. To the best of our knowledge, this is a novel approach in the OD prediction literature.

The rest of the paper is organized as follows. Section II reviews recent literature from a deterministic, rule-based algorithm to a more probabilistic approach, marked by advances in the deep learning framework. Section III describes the dataset and formalizes different ways of defining the OD graphs. Section IV explains the GCN model, formulates the hybrid GCN-MC model, and proposes an evaluation metric for performance evaluation. Next, Section V presents the results and discusses notable findings. Lastly, Section VI concludes the paper by identifying the limitations of the current model and the future research direction.

II. Related Work

From linear-based modeling [7, 8, 9] to a nonlinear approach of nonparametric regression and chaotic models [10, 11], OD prediction is now leveraging deep learning framework to make use of the available big data [12, 13]. In particular, GCN has consistently performed better on network-based prediction tasks, including that of OD trips, because the model can capture the non-Euclidean characteristic and spatio-temporal dependency of the dataset [14].

Other studies are more focused on developing a better OD data structure rather than the actual modeling. For instance, [15, 16] proposed OD graphs that are sensitive to the underlying spatiotemporal dependency, while [17] combined users' context and their social media captions to extract clues for their next destination.

III. Dataset

A. Smart Card Overview

In Singapore, the Mass Rapid Transit (MRT) system, was constructed by the Land Transport Authority (LTA), and consists of five MRT lines. In addition to the MRT Network, Singapore also has more than 5,000 bus stations scattered across the city-state [18].

EZ-Link card is the smart card used in Singapore for the access of public transportation, including buses and MRT. As described in Table I, each EZ-Link record is associated with a ride, including the boarding and alighting station IDs and its corresponding entry and exit timestamps. Other information such as the travel mode and commuter types are also recorded. In addition, each smart card is associated with an anonymized and encrypted unique identifier to protect commuters' privacy [6].

In our study, we analyze the data captured by the EZ-Link card from January 1st 2016 to May 31st 2016, corresponding to 70,000+ randomly selected commuters. There are around 7.5 million OD trips made by those 70,000+ commuters. In addition, we primarily focus on bus trips, rather than MRT trips. This is because the former offers greater OD trips granularity, and therefore, a more nuanced mobility behavior.

TABLE I: Smart Card Dataset Attributes

Attribute	Description
card id	unique identifier of a smart card
type	commuter type (i.e., child, adult, senior)
mode	transport mode of the ride
entry date	starting date of a ride
exit date	ending date of a ride
entry time	starting time of a ride
exit time	ending time of a ride
origin id	unique identifier of the origin MRT station/bus station
destination id	unique identifier of the destination MRT station/bus station

B. Trips Regularity

An important feature of public commuting, particularly in Singapore, is its *regularity*. When surveying approximately $200,000$ randomly selected users across the 5-month period, Figure 1 illustrates how the number of users exponentially falls with an increasing number of unique final destination This insight is significant in our problem formulation, especially when we try to combine GCN with an MC algorithm (to be detailed in Section IV-C). The hybrid GCN-MC will penalize incorrect next station prediction, depending on how likely the next state is in relation to the global-level probability distribution.

Overall, the regularity of trips is commonly observed in public transport and ride-sharing services, including that of bicycle and e-scooters [19, 20, 21, 22]. On the other hand, social-media-related trips, such as those found in Foursquare, are generally more irregular [23].

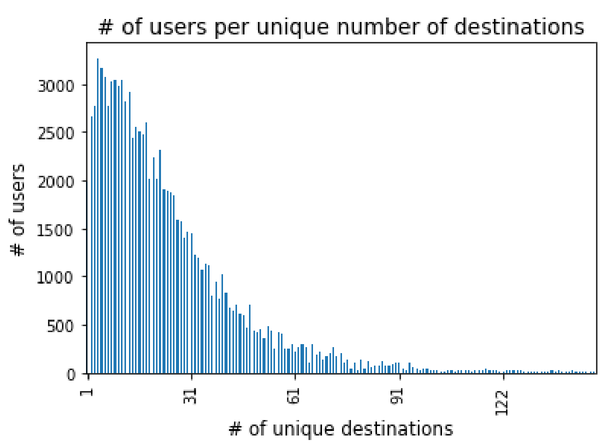


Fig. 1: Regularity of Trips in Singapore's Public Transportation Dataset (based on $200,000$ randomly sampled commuters with a total of 21.3 million trips)

C. OD Graph Definition

For this study, a collection of *daily* trips made by *one* user (i.e., a commuter) is represented as a directed graph, g.

$$g = (\boldsymbol{\mathcal{V}_g}, \boldsymbol{E_g}, \boldsymbol{A_g}), \quad g \in \mathcal{G} \tag{1}$$

Here, $\mathcal{G}$ is the entire collection of directed graphs corresponding to a given set of users; $\boldsymbol{\mathcal{V}_g}$ is a set of nodes in a graph g, representing different bus stations, and $|\boldsymbol{\mathcal{V}_g}|$ is the total number of nodes; $\boldsymbol{E}_g$ is a set of edges in graph g referring to the connectivity between nodes. An edge, $e(v_1, v_2) \in \boldsymbol{E}_g$, implies that there is at least one trip made by the user from v_1 to v_2. In some of the OD graphs, each edge $e(v_1, v_2) \in \boldsymbol{E}_g$ may carry a weight, e.g. the travel duration for a trip from v_1 to v_2. $\boldsymbol{A_g}$ = $(\boldsymbol{a}_{ij})_{|\boldsymbol{\mathcal{V}_g}|\times|\boldsymbol{\mathcal{V}_g}|} \in \mathbb{R}^{|\boldsymbol{\mathcal{V}_g}|\times|\boldsymbol{\mathcal{V}_g}|}$ is the adjacency matrix that captures the graph's connectivity information. In other words, $\forall \boldsymbol{a}_{ij} \in \boldsymbol{A_g}$, if $\exists e(v_i, v_j) \in \boldsymbol{E_g}$, $\boldsymbol{a}_{ij} = 1$; otherwise, $\boldsymbol{a}_{ij} = 0$.

In addition, we maintain a feature matrix $\boldsymbol{X_g^{v_i}}$ for each node v_i in graph g, where $v_i \in \boldsymbol{\mathcal{V}_g}$ and $g \in \mathcal{G}$. To be more specific, $\boldsymbol{X_g^{v_i}} = [X_1^{v_i}, \cdots, X_{|F|}^{v_i}] \in \mathbb{R}^{1\times|F|}$. The entire feature space, F, represents a variety of spatiotemporal attributes, such as the origin location, duration before the next trip, *etc*.

Last but not least, $\boldsymbol{Y_g} = [Y_1, Y_2, \cdots, Y_{|\mathcal{V}_g|}] \in \mathbb{R}^{1\times|\mathcal{V}_g|}$ is the target vector of a graph g with $|\mathcal{V}_g|$ nodes. The target variable, Y_{v_i}, of a node v_i, is the node's next station unique identifier (ID). The value for Y_{v_i} can either be the destination ID if v_i is the originating station or the next-origin ID if v_i is the alighting station, but not the terminal station[1]. After completing the feed-forward operations as discussed in Section IV-B, a node v_i is going to output a prediction matrix of size $(1 \times n_d)$, where n_d refers to the total number of next station predicted. The model then selects the next station prediction, $\hat{y}_{v_i}$, with the highest confidence level. A comparison between $\hat{y}_{v_i}$ and $y_{v_i} \in \boldsymbol{Y_g}$ is made to calculate the loss and backpropagate the parameter updates through gradient descent.

This paper will consider two different family of graphs: (1) *Spatial-only* and *SpatialTemporal* graphs to evaluate the effects of temporality, as well as (2) *Same-users* and *Random-users* graphs to examine the influence of user's context or patterns on accuracy. In summary, there are a total of four graphs being considered, including the i) *Spatial-only+Same-users*; ii) *SpatialTemporal+Same-users*; iii) *Spatial-only+Random-users*; and iv) *SpatialTemporal+Random-users*.

Spatial-only graphs. For each node in graph g, the spatial feature space represents the originating station ID, while the edge attribute takes the Boolean value of 1 for connection or 0 otherwise.

SpatialTemporal graphs. This class of graphs captures additional temporal information in their *nodes* including:

- *Travel gap*. The duration, in *seconds*, for a user to make the next trip after alighting. For the first and the last trip of the day, the corresponding travel gap is assigned a value of -999.

[1] A terminal node, $v_{terminal}$, represents the final station a user alights to on a particular day, and its $Y_{terminal}$ is assigned the integer value of -999.

- *Boarding hour*. The hour [0..23] by which a trip between two nodes started.
- *Alighting hour*. The hour [0..23] by which a trip between two nodes ended.
- *Is weekday*. A Boolean value indicating whether a trip is made on weekdays.

and in their *edges*, $e(v_i, v_j)$:

- *Trip duration*. The duration of a trip, in *seconds*, between two connected nodes.

Same-users graphs. $70,000$ commuters are randomly selected from the January 2016 dataset, and their subsequent 4-month trips are retrieved (a total of 7.5M records).

Random-users graphs. $70,000$ commuters and their corresponding monthly trips are sampled *for each* of the 5-month period.

The nodes are split into three disjoint subsets, namely training, validation, and testing sets, based on a ratio of 8:1:1. During any one of the three phases, the nodes that are irrelevant to the current phase will be *masked*, e.g. testing and validation nodes are masked during the training phase. Masking allows graphs to preserve their critical adjacency information while maintaining separation across the different phases.

In order to better understand the sizes of graphs generated, we summarize their characteristics and the distribution of testing nodes in Table II. We have made three main observations. Firstly, the number of nodes in each graph is generally small, e.g., over 58% of graphs have up to four nodes. Secondly, the graphs with four nodes are the most common. This represents our everyday commute to work, school, and other places of interests. Lastly, although rare, there are some graphs with more than 14 nodes (about 0.7% of them).

TABLE II: Summary Statistics of Nodes and Graphs Used For Testing

# of nodes per graph	# of graphs	Proportion of testing nodes (%)
2	78,537	7.86
3	30,394	4.61
4	129,159	25.91
5	36,982	9.33
6	47,001	14.13
7	28,398	9.96
8	24,989	10.02
9	11,726	5.28
10	8,757	4.33
11	5,146	2.85
12	3,369	2.05
13	1,894	1.24
>= 14	3,067	2.42

IV. Methodologies

A. Markov Chain Baseline

Markov Chain (MC) is a stochastic model that describes the probability of a next state, given its current state and the probability of transition [24, 25]. In our study, the MC model calculates the proportion of correct (i.e. within the top-k, where $k = 1$ and $k = 3$ in our implementation) next state predictions, $v_i.next$, given its current state, $v_i.current$. Its pseudo-code is defined in Algorithm 1.

Algorithm 1 Baseline Markov-Chain Model Algorithm (for Top-k)

```
Input: HashMap_freq ← ⟨origin : next_1, ··· , next_k⟩,
graph collection 𝒢
Output: accuracy
total_correct ← 0; total_nodes ← 0; accuracy ← 0
foreach g ∈ 𝒢 do
    foreach v ∈ 𝒱_g^test do
        if ∃ v.next and v.next ∈ HashMap_freq[v.current] then
            total_correct ← total_correct + 1
        total_nodes ← total_nodes + 1
accuracy ← total_correct/total_nodes
returns accuracy
```

$HashMap_{freq}$ is generated from the global OD trips dataset. The key represents a collection of originating station IDs and its corresponding value represents a list of top-k next station IDs.

B. Graph Convolutional Network (GCN) Baseline

GCN is commonly used to solve node embedding, link prediction, and node classification problems, such as OD prediction [26]. Different from its Graph Neural Network (GNN) predecessor, the convolution layer in GCN learns the latent representations of neighboring nodes in a *Message Passing (MP)* process. MP dictates that for each node, $v_i \in \mathcal{V}_g$, an aggregation function will combine and transform its adjacent nodes' embeddings at time t to yield a new node embedding $X_g^{v_i}$ at time $t+1$. For our study, we use a linear aggregation function [26], as defined in Equation (2).

$$X_g^{v_i}(t+1) = D^{-0.5} A D^{-0.5} X_g^{v_i}(t) \cdot \Theta \tag{2}$$

where, D is the degree matrix, A is the adjacency matrix, and Θ is a set of learnable weights and bias matrices.

The transformed feature matrix is then passed through a post-MP module, which consists of two fully linear networks (dropout rate = 0.25) with an output size of 1024 and n_d respectively. The working diagram of the complete GCN model is illustrated in Figure 2.

We use a negative log likelihood loss function with a log softmax non-linear activation (or the *Categorical Cross Entropy (CCE)* as defined in Equation (3)). Here, M is the total number of batched nodes currently evaluated, C is the entire set of possible next stations, y is the target variable, and $\hat{y}$ is the predicted outcome.

$$CCE = -\frac{1}{M} \sum^{M} \sum_{c=0}^{C} y_c \cdot \log \hat{y} + (1 - y_c) \cdot \log(1 - \hat{y_c}) \tag{3}$$

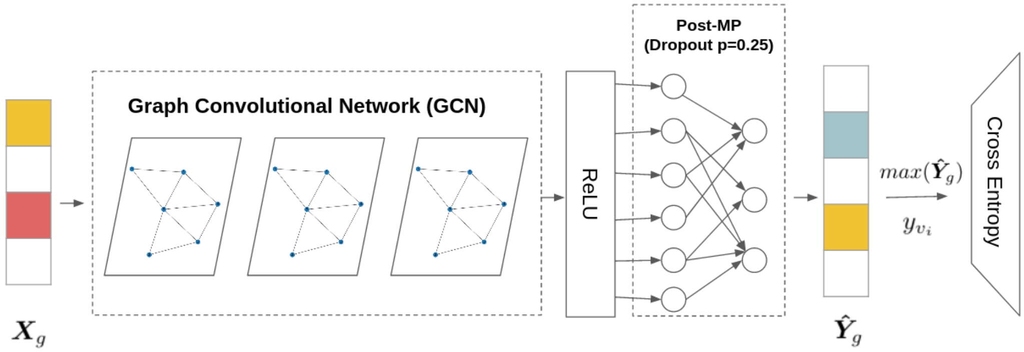


Fig. 2: Our deep graph network with three layers of GCN, a post-MP module consisting of two linear layers, a log softmax activation, and a negative log likelihood loss function.

We optimize the hyperparameters for the final model configuration by conducting a grid scan with a k-fold cross validation approach (where $k = 10$) [27].

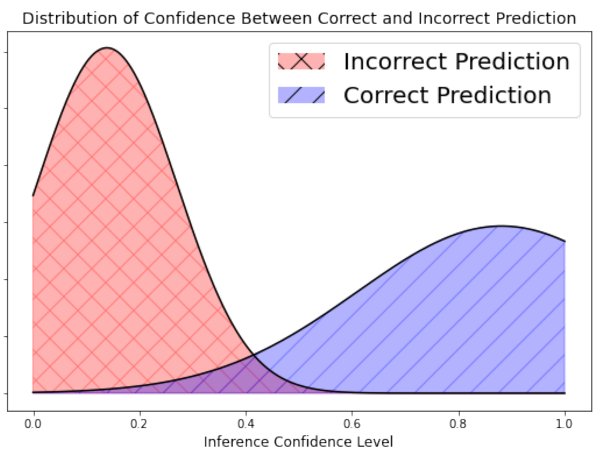


Fig. 3: The Distribution of Confidence Level for Incorrect and Correct Predictions

C. GCN-Markov Chain Hybrid Model

In the hybrid model, we attempt to constrain any uncertain, and often incorrect, prediction to be as close as possible to the general trend. To this end, we incorporated an MC process into the negative log likelihood loss function. Given the regularity of OD public trips as elaborated in Section III-B, the additional penalty is proportional to the inverse of $p(\hat{y}_i|x)$, where x is the current state and $\hat{y}_i$ is the predicted next state. Essentially, the factor proportionately penalizes *incorrect* prediction that occurs infrequently in the global distribution of OD trips as described in Equation (4).

$$MarkovChain\_CCE = (1 + \mathit{PenalizingFactor}) \cdot CCE \quad (4)$$

where the *PenalizingFactor*, $PF(X_g.current, \hat{Y})$, takes in as input a set of current stations $X_g.current$, and their corresponding predictions $\hat{Y}_g$, to return a single continuous scalar value that lies within $[0, 1]$ by applying a Sigmoid function as in Equation (5).

$$PF = Sigmoid\left(\sum_{i=1}^{|V_g|} h(y_{v_i}, \hat{y}_{v_i}) \cdot \frac{1}{p(\hat{y}_{v_i}|x_{v_i}.current)}\right) \quad (5)$$

The boolean function, $h(y_i, \hat{y}_i)$, ensures that the modified loss function applies only to nodes that make incorrect prediction as in Equation (6).

$$h(y_{v_i}, \hat{y}_{v_i}) = \begin{cases} 1, & \text{if } y_{v_i} \neq \hat{y}_{v_i} \\ 0, & \text{if } y_{v_i} = \hat{y}_{v_i} \end{cases} \quad (6)$$

D. Evaluation Metric: Accuracy/Proportion Trade-off

As illustrated in Figure 3, there are two distinct confidence distributions for the incorrect and correct predictions. As expected, the average confidence level is generally higher for correct predictions at 0.98 than for incorrect predictions at 0.22.

These trends beg the question: *what if we predictions only when their confidence is relatively high (ie. above a certain threshold)*? Accordingly, we introduce a confidence threshold, where a lower value will increase the proportion of test cases at the expense of accuracy, and *vice versa*; a *trade-off*.

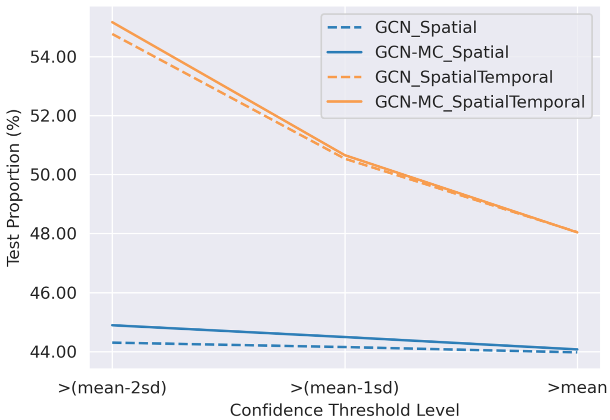
This trade-off metric, as defined in Equation (7), closely resembles the $F1$ score that calculates the harmonic mean between accuracy and proportion.

$$\mathit{Trade\text{-}off\ Score} = 2 \times \frac{accuracy \times proportion}{accuracy + proportion} \quad (7)$$

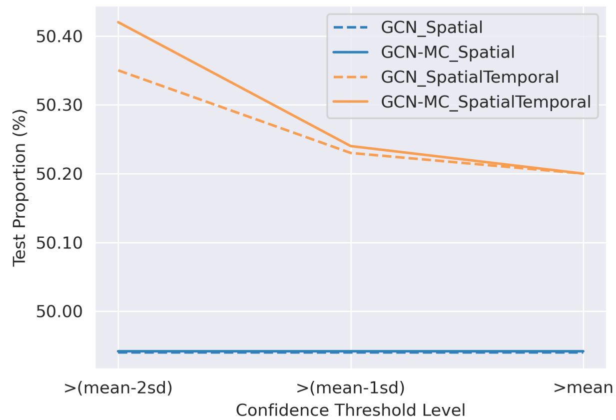
V. Results and Discussion

A. Preliminary Results

In our preliminary evaluation, we compare the performance of the different models and graph data structures with the baseline MC algorithm *without* the dynamic confidence thresholding as summarized in Table III.

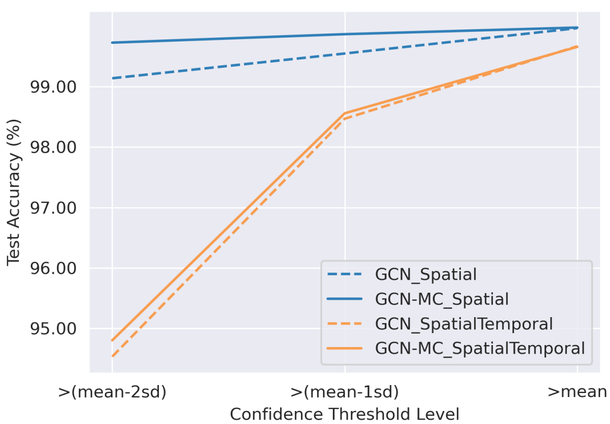


(a) Same-users

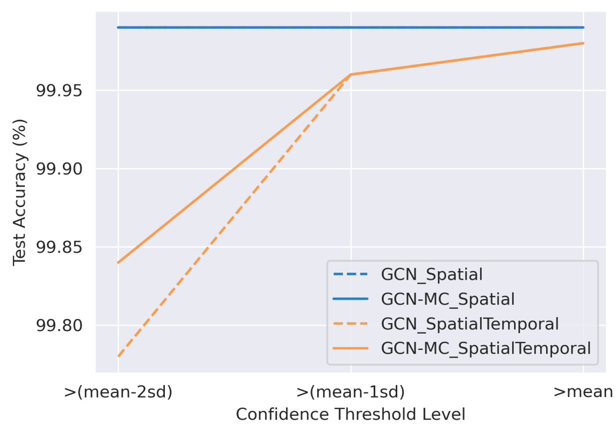


(b) Random-users

Fig. 4: Changes in Proportion (%) with Increasing Confidence Threshold for (a) Same-users, and (b) Random-users



(a) Same-users



(b) Random-users

Fig. 5: Changes in Accuracy (%) with Increasing Confidence Threshold for (a) Same-users, and (b) Random-users

TABLE III: Preliminary evaluation *without* the dynamic thresholding of the confidence level

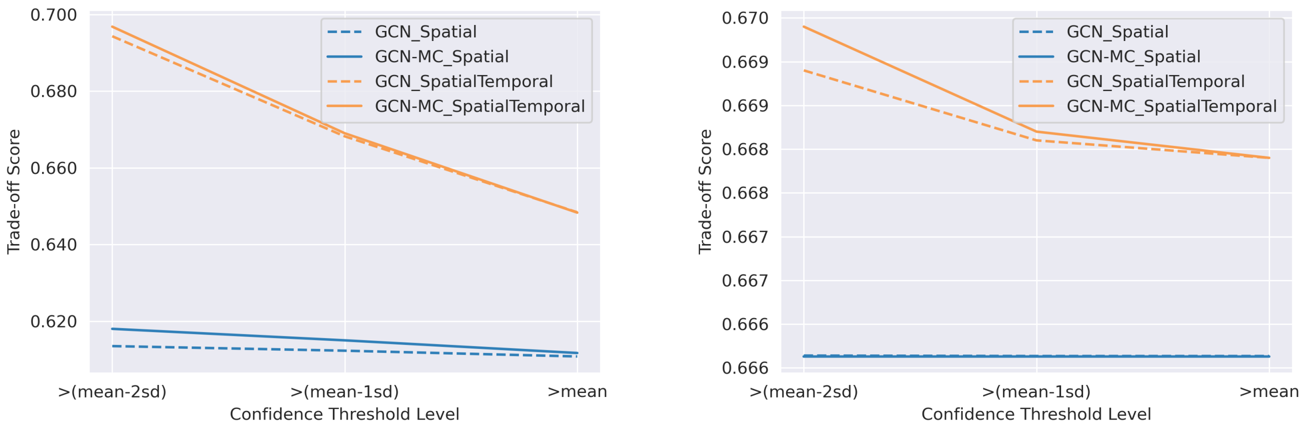
Model Types	Accuracy (%)	
	Same-users	Random-users
MC_Spatial(Top-1)	25.08 ± 0.01	26.10 ± 0.03
MC_SpatialTemporal(Top-1)	30.52 ± 0.01	29.25 ± 0.02
MC_Spatial(Top-3)	32.08 ± 0.01	33.10 ± 0.03
MC_SpatialTemporal(Top-3)	40.52 ± 0.01	39.25 ± 0.02
GCN_Spatial	42.73 ± 0.01	43.98 ± 0.02
GCN_SpatialTemporal	48.40 ± 0.07	47.48 ± 0.20
GCN-MC_Spatial	$\mathbf{57.07 \pm 0.01}$	$\mathbf{58.86 \pm 0.01}$
GCN-MC_SpatialTemporal	$\mathbf{58.91 \pm 0.01}$	$\mathbf{57.43 \pm 0.01}$

GCN significantly improves the baseline MC model by 17% and 18% in the *Spatial-only* and *SpatialTemporal* graphs respectively. The hybrid GCN-MC model further improves GCN by 15% and 11% in the *Spatial-only* and *SpatialTemporal* graphs respectively. If we compare between *Same-users* and *Random-users* graphs, the former is performing better by 1.2% when temporality is considered. This suggests the importance of users' context and temporality in OD prediction tasks.

B. Models' Accuracy versus Proportion Trade-off

The accuracy versus proportion trade-off is computed at every testing epoch by first deriving the confidence distribution of correct predictions and varying the confidence threshold.

Proportion. As illustrated in Figure 4, the hybrid GCN-MC models increase the proportion of confident test cases by $1\% - 2\%$ as compared to their GCN counterparts. In addition, the *SpatialTemporal* graphs perform more predictions than the *Spatial-only* graphs by $1\% - 8\%$. Similarly, the *Same-users* graphs are performing more predictions than the *Random-*



(a) Same-users

(b) Random-users

Fig. 6: Changes in Trade-off Score (%) with Increasing Confidence Threshold for (a) Same-users, and (b) Random-users. The higher the score is the better the model becomes.

users graphs by 4%, particularly when the *SpatialTemporal* graphs are used.

Accuracy. As illustrated in Figure 5, the hybrid GCN-MC models improve the accuracy of their GCN counterparts by $1\% - 1.5\%$. In addition, the *Spatial-only* graphs demonstrate higher accuracy than the *SpatialTemporal* configuration by $1\% - 5\%$. The *Spatial-only* graphs also yield consistently high accuracy ($> 98\%$) at varying level of confidence threshold, while the accuracy for *SpatialTemporal* graphs fall significantly to $90\% - 95\%$ when the confidence threshold is dropped beyond one standard deviation about the mean of the confidence distribution. Moreover, *Same-users* graphs produce lower accuracy than *Random-users* by 1% and 5% for *Spatial-only* and *SpatialTemporal* graphs respectively.

C. Models' Overall Evaluation

As shown in Figure 4 and Figure 5, a drop in confidence threshold will increase the proportion of evaluated nodes at the expense of accuracy. When we combine both the accuracy and proportion into a single trade-off score defined in Equation (7), the performance of the different models and graph structures can be evaluated objectively. As illustrated in Figure 6, the hybrid GCN-MC models improve the overall performance of their GCN counterparts by $2\% - 4\%$. In addition, the *SpatialTemporal* graphs are performing better than the *Spatial-only* graphs by $2\% - 12\%$. Similarly, trips from *Same-users* perform better than those of *Random-users* by 2.9% when the *SpatialTemporal* graphs are used. This trend suggests that adding both contextual and temporal information is crucial for better OD predictions.

VI. Conclusion

In this paper, we have proposed a novel deep learning architecture to predict OD trips using a large-scale public transportation dataset in Singapore. First, we demonstrated how GCN increases the prediction accuracy of the baseline MC algorithm by $17\% - 18\%$ while the hybrid GCN-MC model further improves the performance of GCN by another $11\% - 14\%$.

Furthermore, we formulated two different approaches to building graphs in the hope of improving the explainability of the deep network. The first approach examined the effect of temporality by constructing *Spatial-only* and *SpatialTemporal* graphs. The second approach focused on the effect of users' context by defining *Same-users* and *Random-users* graphs. We found that both temporality and contextual information improve the overall performance of the models. The hybrid integration of MC algorithm further reinforced the notion of context into GCN, allowing the deep model to be aware of the global trend when faced with an uncertain prediction task.

One limitation of a GCN-based model is its difficulty to train without, to some extent, processing the validation/testing nodes either through its adjacency or structural information (i.e., transductive learning). In other words, in cases where new sets of training graphs become available, the GCN model will lose its generalization and need to be retrained.

Acknowledgments

This research is supported by the Ministry of Education, Singapore, under its AcRF Tier 2 Funding (Grant No: MOE2019-T2-2-116). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the Ministry of Education, Singapore.

References

[1] Lingbo Liu, Jingwen Chen, Hefeng Wu, Jiajie Zhen, Guanbin Li, and Liang Lin. "Physical-Virtual Collaboration Modeling for Intra-and Inter-Station Metro Ridership Prediction". In: *IEEE Transactions on Intelligent Transportation Systems (TITS)* (2020).

[2] Yazhe Wang, Chih-Chieh Hung, Baihua Zheng, and Ee-Peng Lim. "TripDecoder: Inferring routes of passengers of mass rapid transit systems by smart card transaction data". In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 655–658.

[3] Tianming Zhang, Yunjun Gao, Baihua Zheng, Lu Chen, Shiting Wen, and Wei Guo. "Towards distributed node similarity search on graphs". In: *World Wide Web* 23.6 (2020), pp. 3025–3053.

[4] Marc Miska, Hiroshi Warita, and Masao Kuwahara. "Analysis of Tokyo Metropolitan Expressway's demand using ETC-OD data". In: *Proceedings of Infrastructure Planning in Japan* (2009), p. 39.

[5] WANG Yuanqing, YANG Liu, GENG Yin, and Mengmeng Zheng. "OD matrix estimation for urban expressway". In: *Journal of Transportation Systems Engineering and Information Technology* 10.2 (2010), pp. 83–87.

[6] Xiancai Tian and Baihua Zheng. "Using smart card data to model commuters' responses upon unexpected train delays". In: *IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 831–840.

[7] Yasuo Asakura, Eiji Hato, and Masuo Kashiwadani. "Origin-destination matrices estimation model using automatic vehicle identification data and its application to the Han-Shin expressway network". In: *Transportation* 27.4 (2000), pp. 419–438.

[8] Yun-tao Chang, Guo-xiong Peng, and Xiao-guang Yang. "Freeway OD matrices estimation with on/off ramps traffic counts [J]". In: *Journal of Traffic and Transportation Engineering* 4 (2003), pp. 89–94.

[9] Hai-yan Wang, Qiao-jun Xiang, Jian Lu, Cun-hou Gao, and Chen-guang Liang. "Method of expressway ramp OD matrix calculation". In: *Journal of highway and transportation research and development* 22.2 (2005), pp. 108–115.

[10] Jianjun Wang, Gang Wei, Fashun Yang, and Qianqian He. "Expressway OD Matrix Estimation Based on the Fuzzy Optimization Grey Prediction Model". In: *The second International Workshop on Intelligent Systems and Applications*. IEEE. 2010, pp. 1–5.

[11] Wen Du, Daibin Zhao, and Yuan Liu. "Calculation of the Expressway OD Matrix Based on Grey Markov Chain Model". In: *International Conference on Transportation Engineering*. 2009, pp. 2707–2712.

[12] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. "Modeling trajectories with recurrent neural networks". In: *the Twenty-Six International Joint Conference on Artificial Intelligence (IJCAL)*. 2017, pp. 3083–3090.

[13] Hanyuan Zhang, Hao Wu, Weiwei Sun, and Baihua Zheng. "Deeptravel: a neural network based travel time estimation model with auxiliary supervision". In: *the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAL)*. 2018, pp. 3655–3661.

[14] Kyungeun Lee and Wonjong Rhee. "Graph Convolutional Modules for Traffic Forecasting". In: *CoRR* abs/1905.12256 (2019). arXiv: 1905.12256. URL: http://arxiv.org/abs/1905.12256.

[15] Hao Peng et al. "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting". In: *Information Sciences* 521 (2020), pp. 277–290.

[16] Shen Fang, Qi Zhang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. "GSTNet: Global Spatial-Temporal Network for Traffic Flow Prediction". In: *The Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, pp. 2286–2293.

[17] Xi Xiong, Kaan Ozbay, Li Jin, and Chen Feng. *Dynamic Origin-Destination Matrix Prediction with Line Graph Neural Networks and Kalman Filter*. 2020. arXiv: 1905.00406 [cs.LG].

[18] Christopher Tan. *Bus, rail ridership soars to new high*. URL: https://www.straitstimes.com/singapore/transport/bus-rail-ridership-soars-to-new-high.

[19] Xiancai Tian and Baihua Zheng. "Study Group Travel Behaviour Patterns From Large-Scale Smart Card Data". In: *IEEE International Conference on Big Data (Big Data)*. 2019, pp. 1232–1237.

[20] Yazhe Wang, Baihua Zheng, and Ee-Peng Lim. "Understanding the effects of taxi ride-sharing—A case study of Singapore". In: *Computers, Environment and Urban Systems* 69 (2018), pp. 124–132.

[21] Bingjie He, Shukai Li, Chen Zhang, Baihua Zheng, and Fugee Tsung. "Holistic Prediction for Public Transport Crowd Flows: A Spatio Dynamic Graph Network Approach". In: *ECML/PKDD*. 2021, pp. 321–336.

[22] Xiancai Tian, Baihua Zheng, Yazhe Wang, Hsiao-Ting Huang, and Chih-Chieh Hung. "Origin-destination estimation for non-commuting trips using location-based social networking data". In: *Transactions on Data Science (TDS)* 2.3 (2021), 16:1–16:21.

[23] Fan Yang, Peter J Jin, Yang Cheng, Jian Zhang, and Bin Ran. "Origin-destination estimation for non-commuting trips using location-based social networking data". In: *International Journal of Sustainable Transportation* 9.8 (2015), pp. 551–564.

[24] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[25] Luke Tierney. "Markov chains for exploring posterior distributions". In: *the Annals of Statistics* (1994), pp. 1701–1728.

[26] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[27] Sebastian Raschka. *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. 2020. arXiv: 1811.12808 [cs.LG].