Transportation Research Record

A spatial-temporal Deep Learning Framework for Network-wide Bus Passenger Flow prediction

--Manuscript Draft--

Full Title:	A spatial-temporal Deep Learning Framework for Network-wide Bus Passenger Flow prediction
Abstract:	Bus system is one of the most important components of an integrated public transit system that shares a large proportion of daily trips. The accurate short-term bus passenger flow prediction is of great significance for public agencies and operators to ensure the quality and reliability of the services. In this paper, we propose a novel end-to-end deep learning framework that aims at making multi-step collective prediction for different types of bus passenger flow (boarding and alighting) in a network-wide region level. The proposed framework has a deep structure comprising stacks of spatial-temporal blocks, which can effectively learn long-range spatial-temporal dependencies. The large quantities of real-world data from Singapore are used to evaluate the proposed model, according to which we find that our proposed model is capable of making reliable prediction and outperforms a set of commonly used baselines.
Manuscript Classifications:	Data and Information Technology; Urban Tranportation Big Data ABJ30SC; Big Data; Artificial Intelligence and Advanced Computing Applications ABJ70; Artificial Intelligence; Machine Learning (Artificial Intelligence); Neural Networks; Transformative Transit Data AP000; Smartcard; Transportation Issues in Major Cities and Urban Mobility ABE30; Urban Mobility; Public Transportation; Bus Transit Systems AP050; General
Manuscript Number:	
Article Type:	Presentation
Order of Authors:	Siyu Hao
	Dingyi Zhuang
	Der-Horng Lee

A Spatial-temporal Deep Learning Framework for Network-wide Bus Passenger Flow Prediction

Siyu Hao
(Corresponding author)
Department of Civil & Environmental Engineering,
National Unversity of Singapore
Singapore 117576, Singapore
Email: siyuhao@u.nus.edu

Dingyi Zhuang
Department of Civil Engineering and Applied Mechanics
McGill University
Montreal, Quebec H3A 0C3, Canada
Email: dingyi.zhuang@mail.mcgill.ca

Der-Horng Lee
Department of Civil & Environmental Engineering,
National Unversity of Singapore
Singapore 117576, Singapore
Email: dhl@nus.edu.sg

Word count: 4493 words + 1 table $\times$ 250 + 7 figures = 4743 words

Submission Date: 1st Aug 2019

ABSTRACT

Bus system is one of the most important components of an integrated public transit system that shares a large proportion of daily trips. The accurate short-term bus passenger flow prediction is of great significance for public agencies and operators to ensure the quality and reliability of the services. In this paper, we propose a novel end-to-end deep learning framework that aims at making multi-step collective prediction for different types of bus passenger flow (boarding and alighting) in a network-wide region level. The proposed framework has a deep structure comprising stacks of spatial-temporal blocks, which can effectively learn long-range spatial-temporal dependencies. The large quantities of real-world data from Singapore are used to evaluate the proposed model, according to which we find that our proposed model is capable of making reliable prediction and outperforms a set of commonly used baselines.

Keywords: public transit, deep learning, bus, passenger flow, convolutional neural network

INTRODUCTION

An integrated public transport system consists of several different modes of mobility service, among which Mass Rapid Transit (MRT) and Bus are the two most predominant components. Take Singapore for instance, Singapore is recognized as a top-ranked city in terms of public transport system where the trips shared by public transport modes during peak period has already surpassed 63% of total motorized trips. Furthermore, more than 60% of the daily commuters take their public transport trips entirely or partially on buses. Therefore, the bus system forms a critical link for the integrated public transport system, which is incessantly providing indispensable services for a large number of commuters and moving them from place to place. This is why precisely predicting short-term bus passenger flow becomes imperative for relevant agencies and operators to provide better demand-responsive and more reliable service. Thanks to the installment of Automatic Fare Collection (AFC) system on bus, it records massive data containing the exact boarding and alighting activities of each passenger, which provides us with abundant data resources as well as a set of research prerequisites. Accordingly, a number of attempts have been made by researchers from different domains to investigate the bus passenger flow prediction and related tasks. Previously, most of the prediction frameworks are mainly based on either some classical statistical models or the extension of traditional time series methods. For example, Zhang et al. (*1*) predicted the short-term bus passenger flow on certain bus stops using the Kalman-filter model, of which the accuracy was demonstrated to be better than basic Artificial Neural Network (ANN) model. Gong et al. (*2*) proposed a sequential framework to predict short-term bus passenger demand. This sequential framework consists of three stages, where the 1st stage aims at predicting arrival passenger count and in-vehicle empty space count based on the seasonal auto-regressive integrated moving average model (ARIMA), the 2nd stage is designed to predict departure passenger count using a event-based method, and the 3rd stage is to forecast number of waiting passengers at bus top using a Kalman-filter based model. In addition, Ma et al. (*3*) proposed a hybrid method based on Interactive Multiple Model (IMM) to forecast passenger demand on a single bus line. In this particular framework, the statistical pattern models capture different time series patterns firstly and then the IMM algorithm assigns different weights to pattern models according to their prediction performances in order to boost the final prediction accuracy. Similarly, Xue et al. (*4*) also investigated the IMM based hybrid model, which further demonstrated that this hybrid framework normally outperforms other single models in bus passenger demand prediction. Furthermore, on account of the complexity, uncertainty, and nonlinearity of the short-term passenger flow patterns, some machine learning based approaches also have been utilized in this particular prediction task. Chen et al. (*5*) presented a prediction model based on Least Squares Support Vector Machine (LS-SVM), of which the parameters are optimised by an improved genetic algorithm. Yang and Liu (*6*) combined Affinity Propagation (AP) clustering algorithm and SVM to forecast passenger flow on bus rapid transit. Yu et al. (*7*) predicted bus passenger flow in zone level using an Artificial Neural Network (ANN). However, the traditional machine learning based models still have some limitations in modelling extremely complex and highly nonlinear data, especially for the data in a high dimensional tensor form.

In recent years, with the significant improvement of computation power as well as the acquisition of massive data, it remarkably facilitates the development of deep learning techniques. Since deep learning models are able to learn the intricate structure in large data sets, they have already made some striking breakthroughs in computer vision, speech recognition, natural language processing and many other domains (*8*). Accordingly, many researchers have started to explore

the application of different types of deep learning techniques in transportation domain. Since the recurrent Neural Network (RNN) and its variant Long-short Term Memory (LSTM) based models are good at modelling sequential data and capturing temporal features, they have been used in many transportation prediction tasks, including traffic speed estimation (*9*); (*10*); (*11*)), short-term traffic flow prediction (*12*); (*13*). Additionally, Convolutional Neural Network (CNN) based models can better characterize spatial information through the convolution operations, so they have been applied to some large-scale prediction tasks in transportation domain as well. Ma et al. (*14*) presented a CNN based model that learns traffic as image in order to predict the network-wide traffic speed. Zhang et al. (*15*) proposed a deep CNN based framework with residual connections that can learn long-range spatial dependencies, which showed great accuracy in predicting the traffic flows in each region of a city. In order to better capture the spatial-temporal patterns in modelling transportation related data, some hybrid models with more complex structure have been studied as well, such as using ConvLSTM to predict short-term passenger demand under on-demand ride service platform (*16*), parallely combining CNN and Bi-directional LSTM to predict network-wide ridership in urban metro system (*17*), integrating manually designed features with LSTM to predict metro passenger flow (*18*) as well as using Graph Convolutional Neural Network (GCN) based approach to predict short-term demand in bike-sharing system (*19*).

However, compared with the above-mentioned application scenarios, much less attention have been paid on predicting short-term passenger flow in urban bus system leveraging advanced deep learning techniques. The current limitations and unexplored research gaps in this particular tasks are listed in the following aspects:

- **Prediction Scale** - In real world situations, there are thousands of bus stops distributed over the entire urban area, and normally there are very complex spatial-temporal correlations among passenger flows at different bus stops. However, most of the previous relevant research work only focuses on predicting passenger demand for a single bus stop or few key bus stops along a single bus line, which might limit their applicability and scalability in many real large-scale cases.

- **Prediction Level** - Bus stops are densely distributed, usually with a number of stops in a small region. Moreover, unlike a metro station, the scale and passenger volume of a single bus stop are usually much smaller. Therefore, stop-level prediction (predicting passenger demand on each stop) might not be the most efficient and practical strategy, especially for large-scale bus network.

- **Prediction Horizon** - There is no uniform standard for the definition of "short-term". The prediction horizon of previous research work varies from "hourly" to "daily". However, for real-time traffic management, definitely it would be more helpful if the accurate prediction is made in a finer resolution, such as 15 minutes. In addition, the multi-step ahead prediction would also be better than the single-step prediction.

- **Different types of bus passenger flow** - Many research work predicted bus passenger demand regardless of the different types of passenger flow. However, in a particular time interval, the number of boarding passengers and the number of alighting passengers at the same bus stop could be much different. Therefore, making prediction considering different types of passenger flow is also of great importance.

To tackle the above-mentioned gaps, we present a novel end-to-end deep learning framework to predict the short-term bus passenger flow in network-wide region level. In the proposed framework, a series of targeted adjustments and designs based on Deep Convolutional Neural Network (DeepCNN) have been made to better capture the complex spatial-temporal patterns of different passenger flows in a large-scale bus network. This framework can simultaneously predict both the number of boarding passengers and alighting passengers at each region of the city in the near future. In addition, the framework is evaluated and validated with large quantities of real word data collected from Singapore's AFC system. The remainder of this paper is organized as follows. In section 3, we elaborate all the preliminaries and methodologies in this research work. In section 4, we introduce the experiment settings and analyze the results in depth. Finally in section 5, we summarize our main findings and share the potential future research directions.

METHODOLOGY

In this section, we firstly introduce the definitions of all the relevant concepts and then elaborate the concrete structure of the proposed framework.

Prediction Scope

Network-wide Region Level Prediction

As discussed above, stop-level prediction (predicting passenger demand on each stop) might not be the most efficient and practical strategy when predicting bus passenger flow, especially in a large-scale public transit network. Therefore, in our study, we predict the bus passenger flow in network-wide region level. Specifically, the city area is uniformly partitioned into $I \times J$ grids, and each of the grid represents a region. Then, for all the bus stops within the city area, we assign them to the corresponding region according to their longitude and latitude coordinates, which can be defined as follows:

$$G_{i,j} = \{s \in S | s_c \in (i,j)\} \tag{1}$$

Where S refers to the set of all the bus stops in the city area. s represents a particular bus stop in S and s_c refers to its coordinates. Thus, G_i, j is the collection of all stations that lie in grid (i,j). This indicates how we aggregate several adjacent stops into region level.

Two types of passenger flow

We categorize the bus passenger flow into two different types, namely the boarding flow and alighting flow, which are defined as follows:

$$x_t^{b,i,j} = Card(\{ba \in BA_t | ba_s \in G_{i,j}\}) \tag{2}$$

$$x_t^{a,i,j} = Card(\{aa \in AA | aa_s \in G_{i,j}\}) \tag{3}$$

Where BA and AA refer to the collection of all boarding activities and alighting activities at the time interval t. ba denotes a particular recorded boarding activity (recorded when a passenger taps the card to board a bus) and ba_s refers to the boarding stop of that record. Similarly, aa represents an alighting activity and aa_s refers to the alighting stop. $Card$ represents the cardinality of the set (number of elements in the set). Thus, $x_t^{b,i,j}$ and $x_t^{a,i,j}$ respectively represent the total number of boarding passengers and alighting passengers in grid (i,j) during time interval t. Both $x_t^{b,i,j}$ and $x_t^{a,i,j}$ are 2D matrices.

For a specific time interval t, we concatenate the boarding matrix $x_t^{b,i,j}$ and alighting matrix $x_t^{a,i,j}$ along the first axis to get the mixed flow tensor $X_t \in \mathbb{R}^{2\times \mathrm{I}\times \mathrm{J}}$. X_t can be interpreted as an image with 2 channels where the 1st channel refers to the boarding demand in each region at time interval t and the 2nd channel represents the corresponding alighting demand. The objective of this framework is to predict the the number of boarding passengers and the number of alighting passengers in each region in the near future according to the mixed flow tensors from the last few short-term periods. Thus, the output of the prediction is a tensor denoted by $\hat{Y} \in \mathbb{R}^{2\mathrm{T}'*\mathrm{I}*\mathrm{J}}$, where T' is length of the predicted time intervals in the future ($|t'_1, t'_2, \ldots t'_n|$). The input for the framework is the tensor that stacked by multiple mixed flow tensors of different time periods ($X \in \mathbb{R}^{2\mathrm{T}*\mathrm{I}*\mathrm{J}} = [X_{t1}, X_{t2}, \ldots, X_{tn}]$), where T is the length of the input time intervals.

Modelling the Long-range Dependencies

Convolutional Neural Network

Thanks to the salient characteristics of Convolutional Neural Network (CNN), especially the parameter sharing and the sparsity connection, it enables CNN based models to perform superiorly in a variety of computer vision applications (*20*); (*21*). The output of a convolution operation only depends on few nearby inputs neurons, which also indicates that the convolution operation can effectively capture the local spatial patterns. Thus, a CNN based module is utilized to extract the spatial dependencies among bus passenger flows. Fig 1 illustrates how convolution operations work on the input tensors of passenger flows.

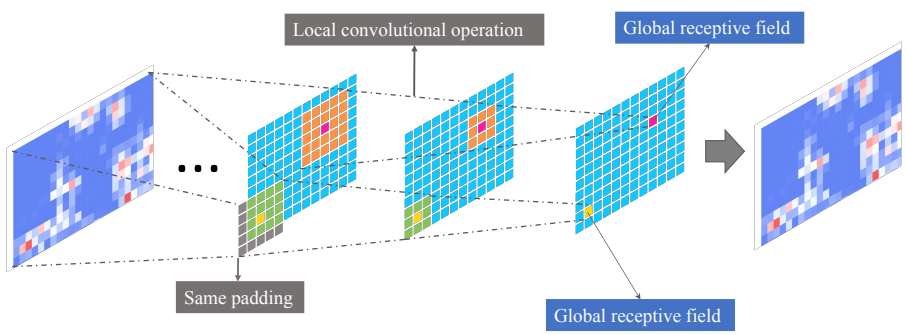


FIGURE 1 : Convolution operations

Deep Residual Network

As shown in Fig 1, in order to guarantee that the input tensor X and output tensor $\hat{Y}$ are of same resolution ($I \times J$), we only adopt consecutive same-padding convolutional layers without any down-sampling operations (e.g. pooling). The receptive field is one of the basic concepts in CNN, which could be defined as the region in the input space that a particular cell of the internal feature map depends on. As shown in Fig 1, stacking multiple convolutional layers can help to increase the receptive field, allowing each cell in the output tensor gather information from a larger area in the input space, which also means that the long-range spatial dependencies could be effectively captured. Concretely, the receptive filed can be calculated as follows:

$$r_l = \begin{cases} k_l & l = 1 \\ r_{l-1} + [(k_l - 1) \times \prod_{i=1}^{l-1} s_i] & l \geq 2 \end{cases} \tag{4}$$

Where r_l is the receptive field at layer l, k_l refers to the kernal size at layer l, s_i denotes the stride size at layer i. In our study, we partition Singapore into grid-like form with shape of 36×21 (input size). Additionally, we fix the kernal size $k = 3 \times 3$ and stride size $s = 1 \times 1$ for all convolutional layers. Hence, according to equation 4, we need to stack at least 17 convolutional layers so that it can capture the spatial correlations between any two particular regions in the city space.

However, the models with very deep structure tend to be difficult to train. Fortunately, He et al. (*22*) proposed a deep residual learning framework that significantly ease the training of deep networks. A deep residual learning framework consists of many residual blocks, thanks to the skip connection mechanism in the residual block, it allows the the activation from layer l to be passed not only layer by layer, but also directly to the deeper layer (e.g. layer $l+2$), of which the whole path is governed by the following equations.

$$a^{(l+1)} = ReLU(W^{(l+1)} \times a^l) \tag{5}$$

$$a^{(l+2)} = ReLU(W^{(l+2)} \times a^{(l+1)} + a^l) \tag{6}$$

Where $W^{(l+1)}$ and $W^{(l+2)}$ refer to the trainable weighted parameters of layer $l+1$ and $l+2$, respectively. a^l denotes the activation from the corresponding layer l and $ReLU$ is the non-linear activation function which is defined in Equation 9.

$$ReLU(x) = max(0, x) \tag{7}$$

Inspired by the advantages of residual learning, we stack multiple residual blocks forming a deep structure in order to model the network-wide spatial correlations. In our study, each residual block has two consecutive weight layers and each of the weigh layer consists of a convolutional layer (kernal size 3×3) followed by a batch normalization layer.

Extracting the Channel-wise Information

Although the deep residual framework can model the long-range spatial dependencies effectively, the channel-wise information is also very important. Hu et al. (*23*) proposed a Squeeze-Excitation structure that can explicitly extract the channel-wise features and improve the performance in a variety of computer vision tasks. Hence, we explore the correlations among channels based on the core idea of SE block. Fig 2 illustrates how the channel information of the passenger flow data could be modelled by SE block.

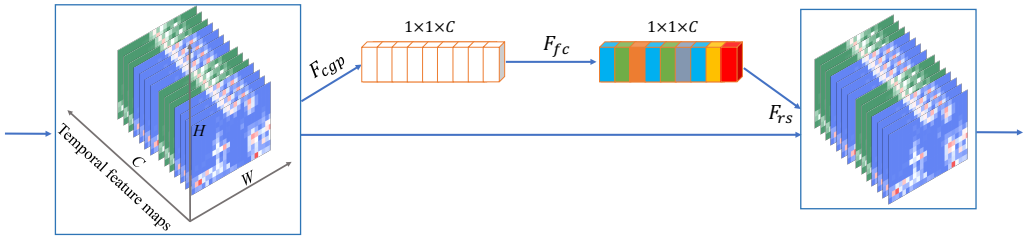


FIGURE 2 : The schema of using SE based block to model the channel-wise information

To clarify, F_{cgp} in Fig 2 refers to the channel-wise global average pooling operation, which is defined as:

$$F_{cgp}(x_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{8}$$

Where $x_c \in \mathbb{R}^{\mathrm{H} \times \mathrm{W}}$ is the spatial feature matrix at the cth channel and $H \times W$ is the size of spatial dimension that equals to $I \times J$ in our study. The output from the channel-wise global average pooling is a vector whose length equals to the number of channels (C) in the input tensor and each element of the vector represents the aggregated information about that channel. Then, the vector is fed into a fully connected feed-forward neural network (denoted by F_{fc}) with sigmoid activation that scales the output into the range of [0,1] (the output vecotr is denoted by s), which can be defined as:

$$s = F_{fc}(z) = \sigma(W_z \times z) \tag{9}$$

Where $z \in \mathbb{R}^{1 \times 1 \times \mathrm{C}}$ is the output from the channel-wise global average pooling operation, W_z refers to the trainable weighted parameters, and σ denotes the sigmoid activation function. Here, s_c (the cth element of vector s) could be interpreted as the importance or attention weights of the spatial feature matrix at cth channel(x_c). Then, we apply the channel-wise multiplication (F_{rs}) between s_c and x_c to obtain the new feature matrix $\tilde{x}_c \in \mathbb{R}^{\mathrm{H} \times \mathrm{W}}$ with channel correlations being embedded in.

As this structure can be easily integrated into standard CNN based architectures (*23*), we then add this block after the second weight layer in the residual block to form an extended residual block, which we term as the *spatial-temporal* block (Fig 3).

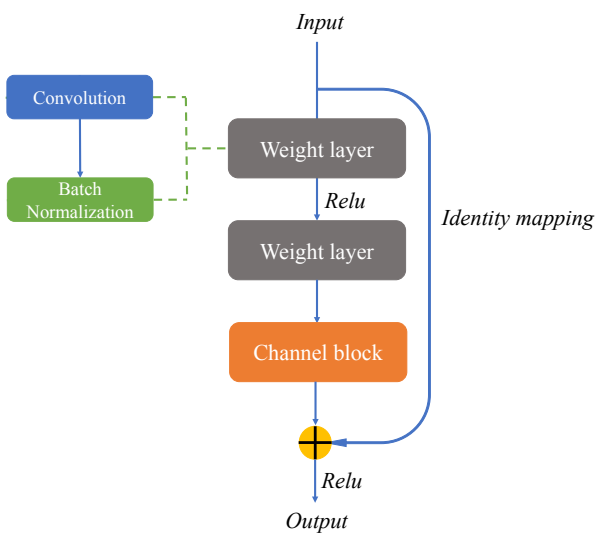


FIGURE 3 : The structure of the *spatial-temporal* block

Fig 4 presents the complete structure of the proposed framework. As illustrated, the input tensor $X \in \mathbb{R}^{2\mathrm{T} \times \mathrm{I} \times \mathrm{J}}$ will be first transformed by a convolutional layer, followed by a series

of *spatial-temporal* blocks, and finally through another convolutional layer that transforms the intermediate outputs into the final output with the required form ($\hat{Y} \in \mathbb{R}^{2T' \times I \times J}$).

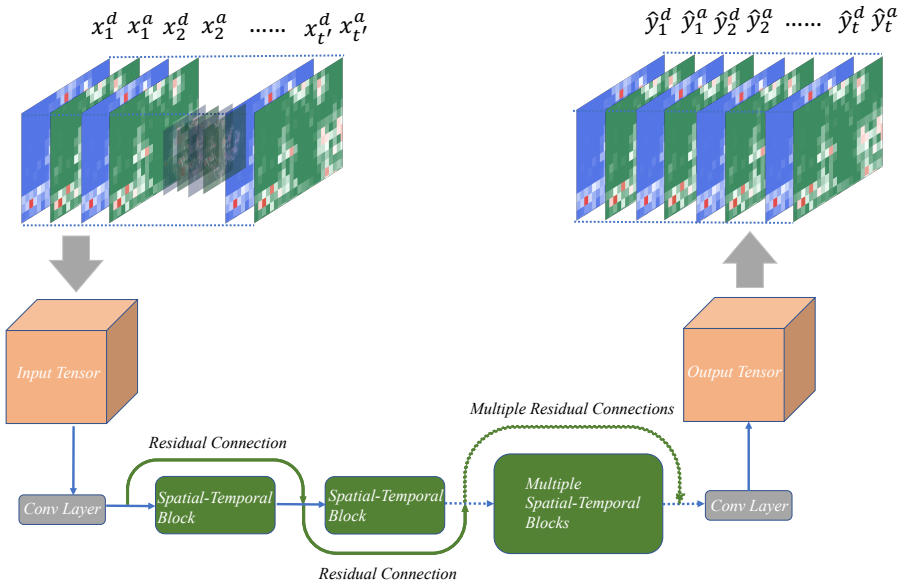


FIGURE 4 : Framework overview

EXPERIMENTS AND RESULTS

In this section, we firstly elaborate the experiment settings and then evaluate the performance of the proposed framework using the real world smart card data collected from Singapore's AFC system.

Data Preparation

The dataset we use in this study covers all the smart card paid bus trips in Singapore during 19th March 2012 to 25th March 2012. We apply the Min-Max normalization (defined as equation 10) to scale the data into the range of [0,1] in order to improve the training efficiency.

$$z = \frac{x - min(x)}{max(x) - min(x)} \tag{10}$$

Training Settings

We perform the randomly split on the data where 85% of the data are used to train the model while another 15% are the testing set. The batch size of the training data is 32. In the training phase, the early-stopping mechanism is applied to monitor the loss on the testing set while the maximum training epochs is fixed to 500 epochs. The Root Mean Square Error (RMSE) is adopted as the evaluation metric, which is defined as:

$$z = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(Y^{(n)} - \hat{Y}^{(n)})^2} \tag{11}$$

Where N is the total number of observations, $\hat{Y} \in \mathbb{R}^{2T' \times I \times J}$ is the predicted flow tensor and $Y \in \mathbb{R}^{2T' \times I \times J}$ is the real flow tensor. We apply the Adam algorithm (*24*) to train the model by minimizing the RMSE between the predicted flow tensor and the real flow tensor. All the models are built in Pytorch (1.1.0) with CUDA (9.0) and the experiments are implemented on our Google Cloud Platform (GCP) server with 2 NVIDIA Tesla T4 GPUs.

Results Comparison

We compare the proposed framework (DeepConv–RCW) with a set of classical models that have been widely used in transportation prediction tasks, including ARIMA, Long-short Term Memory (LSTM), Gated Recurrent Unit (GRU) and Deep Neural Network (DNN). Moreover, the model based on the deep residual framework but without modelling the channel-wise information in the residual unit (termed as DeepConv–R) and the model based on plain deep CNN without residual connections (termed as DeepConv–P) are also put into the comparison. Table 1 presents the comparison of the overall prediction accuracy among all baselines. The best result in each time window is highlighted in bold.

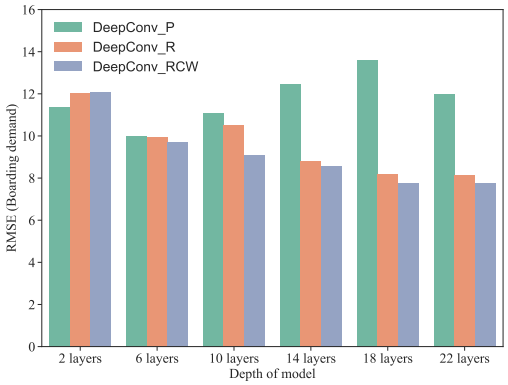
TABLE 1 : Comparisons of the prediction performance (overall)

Model	**0-15mins** (*RMSE*)	**15-30mins** (*RMSE*)	**30-45mins** (*RMSE*)	**45-60mins** (*RMSE*)
ARIMA	14.76	15.12	15.16	15.24
DNN	12.91	12.80	12.76	12.66
LSTM	11.28	10.88	10.78	11.12
Stacked LSTM	10.74	10.61	10.42	10.65
GRU	11.31	11.05	10.88	11.23
Stacked GRU	10.97	10.84	10.71	10.80
DeepConv–P	10.59	11.02	10.87	10.49
DeepConv–R	8.98	9.40	9.53	9.36
DeepConv–RCW	**8.09**	**8.53**	**8.78**	**8.64**

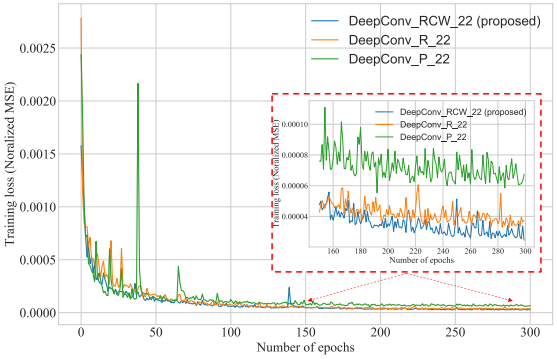
As shown in Table 1, all models in the family of deep learning generate better results than traditional statistical model ARIMA in this particular task. Not surprisingly, RNN based models, such as LSTM and GRU, have better results than DNN, because the temporal correlations could be effectively captured. In addition, stacking multiple layers of LSTM or GRU to form a deeper architecture would also boost the prediction performance while the improvements are marginal. In general, our proposed model DeepConv–RCW outperforms other baselines for all the look-ahead time windows, where the results are 42.1% to 45.2% better than ARIMA, 31.2% to 37.3% better than DNN, 18.6% to 28,3% better than LSTM, 15.7% to 24.7% better than stacked LSTM, 19.3% to 28.4% better than GRU, 18.3% to 26.3% better than stacked GRU, 17.6% to 23.6% better than DeepConv–P and 7.7% to 9.9% better than DeepConv–R.

Impact of the Residual Learning and Channel-wise Learning Mechanism

Fig 5a demonstrates the impact of residual learning and channel-wise learning mechanism for models with different depths.



(a) **Prediction results comparison**



(b) **Learning curves comparison**

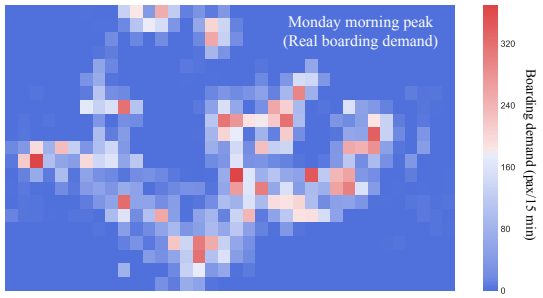
FIGURE 5 : Impact of residual learning and channel-wise learning mechanism

As shown in Fig 5a, in terms of the models based on plain deep CNN without residual learning, denoted by DeepConv–P, the best prediction result occurs when there are totally 6 convolutional layers in the model. As the DeepConv–P becomes deeper, the prediction performance starts to drop and suffer from the vanishing and exploding gradient types of problems. On the contrary, the models with residual learning mechanism (DeepConv–R and DeepConv–RCW) allows them to have very deep architectures without hurting the prediction performance. It is noticed that as the model structure becomes deeper, the gain in prediction accuracy brought by the residual mechanism becomes more and more significant, especially when the depth of the model exceeds 10 convolutional layers. Moreover, the newly designed spatial-temporal block by introducing SE based block, can extract the channel-wise information, which is proved to be very contributive in improving the prediction performance. Additionally, the learning curves for DeepConv–P, DeepConv–R and DeepConv–RCW with same depths (totally 22 convolutional layers) are also plotted in Fig 5b to further highlight the the proposed model's advantages with respect to accuracy as well as consistency.

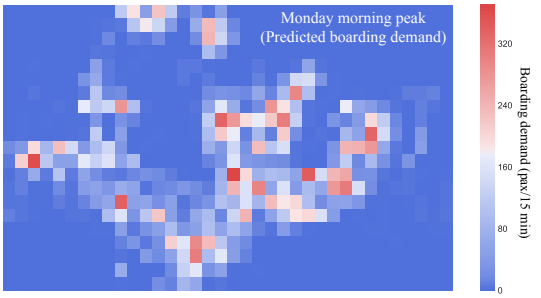
Case Analysis

Here, we visualize several pairs of the real passenger flow and the predicted passenger flow in order to show the prediction results in a more intuitive way. We pick several different scenarios for

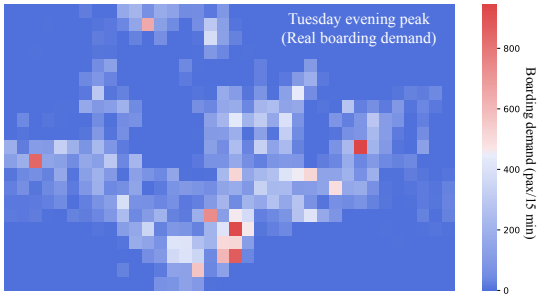
demonstration, including weekday peak periods, holiday eve as well as weekend afternoon.



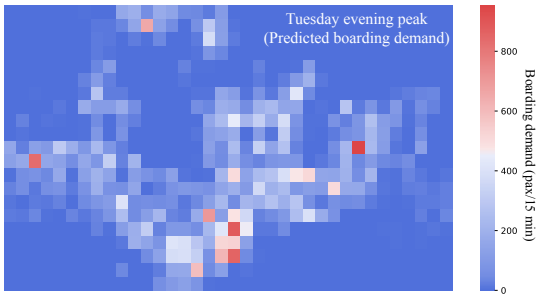
(a) **The real boarding demand (Monday morning peak)**



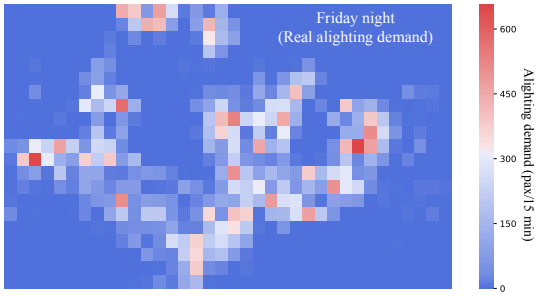
(b) **The predicted boarding demand (Monday morning peak)**



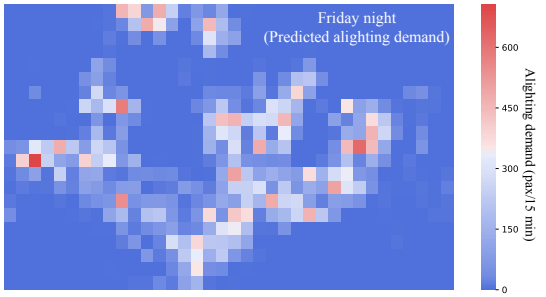
(c) **The real boarding demand (Tuesday evening peak)**



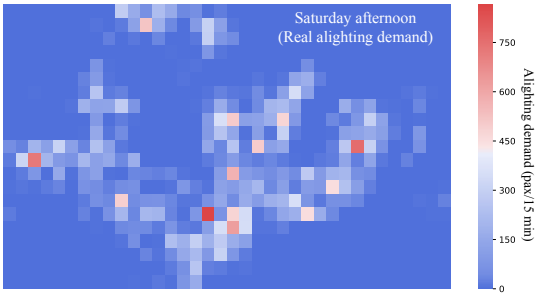
(d) **The predicted boarding demand (Tuesday evening peak)**



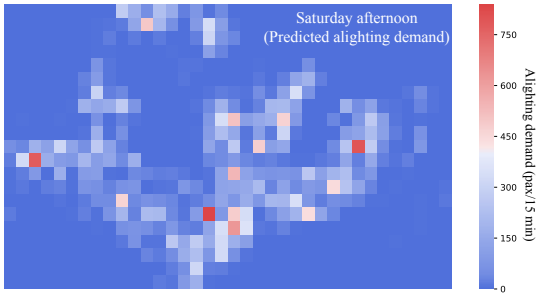
(e) **The real alighting demand (Friday night)**



(f) **The predicted alighting demand (Friday night)**



(g) **The real alighting demand (Saturday afternoon)**



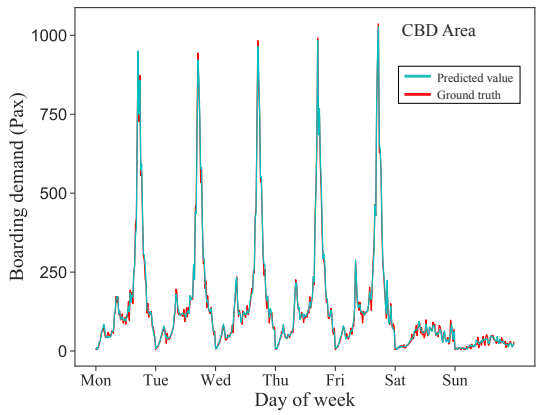
(h) **The real alighting demand (Saturday afternoon)**

FIGURE 6 : Comparisons of the ground truth and predicted passenger flow

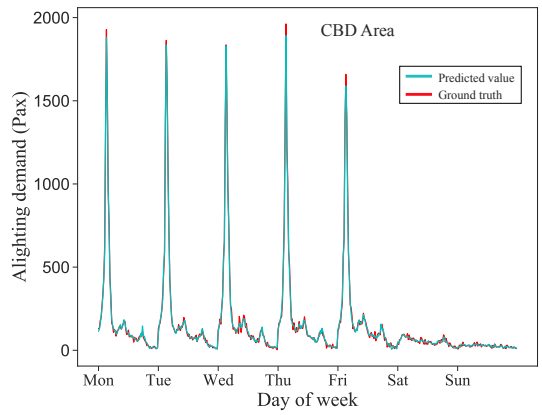
As illustrated in Fig 6, it is noticed that our model is able to make the robust and accurate

prediction for all types of passenger flow (boarding and alighting) under different scenarios.

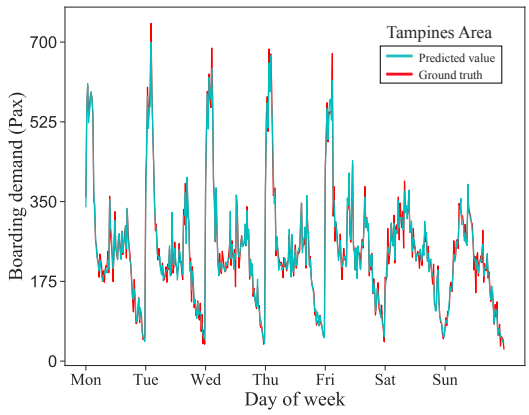
Furthermore, from a relatively more microscopic angle, we select two specific regions with different land use characteristics for evaluation. The 1st selected region is located in CBD area (the grid number is [18,18]) while another region is located in Tampines area covering lots of residential units (the grid number is [9,29]).



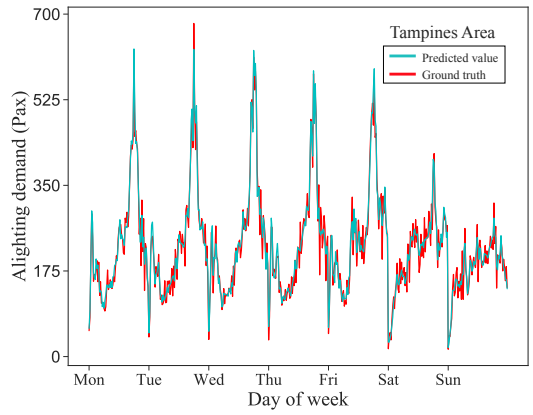
(a) **Boarding demand comparison (CBD area)**



(b) **Alighting demand comparison (CBD area)**



(c) **Boarding demand comparison (Tampines area)**



(d) **Alighting demand comparison (Tampines area)**

FIGURE 7 : Comparisons of the ground truth and predicted passenger flow for selected area

As illustrated in Fig 7, our proposed model shows great prediction accuracy for both of the selected regions, notwithstanding their demand patterns are totally different.

CONCLUSION

In this paper, we propose a novel deep convolutional based framework to predict short-term bus passenger flow in a large scale public transit system. The proposed model is designed to simultaneously predict the boarding demand and alighting demand at each region in the near future by a 15-min interval. Taking advantages of the introduced spatial-temporal block, not only the long-range spatial dependencies could be captured, but also the correlations within channel dimension

could be effectively modelled. We evaluate our proposed model on large quantities of real data collected from Singapore's AFC system and compare the prediction results with a group of state-of-the-art benchmarks, including ARIMA, DNN, several variants of RNN based models as well as CNN based models with different structure, where our proposed model achieves the best result. In the future, the proposed model could be further trained on a larger dataset to improve the generalization ability. In addition, finetuning and testing the model under some special scenarios, such as extreme weather, big events, and large-scale metro shutdown, is also a rewarding direction, by which the practicability and responsiveness of the model could be further enhanced. Moreover, external information from multi-sources, such as land use characteristics and the bus operation information (e.g. timetable, bus type, bus bunching), could be taken into consideration with the intention of expanding the research horizon.

ACKNOWLEDGEMENT

We would like to show our gratitude to Land Transport Authority of Singapore for providing the data.

REFERENCES

[1] Zhang, C.-h., R. Song, and Y. Sun, Kalman filter-based short-term passenger flow forecasting on bus stop. *Journal of transportation systems engineering and information technology*, Vol. 11, No. 4, 2011, pp. 154–159.

[2] Gong, M., X. Fei, Z. H. Wang, and Y. J. Qiu, Sequential framework for short-term passenger flow prediction at bus stop. *Transportation Research Record*, Vol. 2417, No. 1, 2014, pp. 58–66.

[3] Ma, Z., J. Xing, M. Mesbah, and L. Ferreira, Predicting short-term bus passenger demand using a pattern hybrid approach. *Transportation Research Part C: Emerging Technologies*, Vol. 39, 2014, pp. 148–163.

[4] Xue, R., D. J. Sun, and S. Chen, Short-term bus passenger demand prediction based on time series model and interactive multiple model approach. *Discrete Dynamics in Nature and Society*, Vol. 2015, 2015.

[5] Chen, Q., W. Li, and J. Zhao, The use of LS-SVM for short-term passenger flow prediction. *Transport*, Vol. 26, No. 1, 2011, pp. 5–10.

[6] Yang, X. and L. Liu, Short-term passenger flow forecasting on bus station based on affinity propagation and support vector machine. *Journal of Wuhan University of Technology (Transportation Science & Engineering)*, , No. 1, 2016, p. 8.

[7] Yu, S., C. Shang, Y. Yu, S. Zhang, and W. Yu, Prediction of bus passenger trip flow based on artificial neural network. *Advances in Mechanical Engineering*, Vol. 8, No. 10, 2016, p. 1687814016675999.

[8] LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. *nature*, Vol. 521, No. 7553, 2015, p. 436.

[9] Ma, X., Z. Tao, Y. Wang, H. Yu, and Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, Vol. 54, 2015, pp. 187–197.

[10] Cui, Z., R. Ke, and Y. Wang, Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. *CoRR*, Vol. abs/1801.02143, 2018.

[11] Wang, J., R. Chen, and Z. He, Traffic speed prediction for urban transportation network: A path based deep learning approach. *Transportation Research Part C: Emerging Technologies*, Vol. 100, 2019, pp. 372–385.

[12] Tian, Y. and L. Pan, Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, IEEE, 2015, pp. 153–158.

[13] Zhao, Z., W. Chen, X. Wu, P. C. Chen, and J. Liu, LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, Vol. 11, No. 2, 2017, pp. 68–75.

[14] Ma, X., Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, Vol. 17, No. 4, 2017, p. 818.

[15] Zhang, J., Y. Zheng, and D. Qi, Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*, 2017, pp. 1655–1661.

[16] Ke, J., H. Zheng, H. Yang, and X. M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, Vol. 85, 2017, pp. 591–608.

[17] Ma, X., J. Zhang, B. Du, C. Ding, and L. Sun, Parallel Architecture of Convolutional Bi-Directional LSTM Neural Networks for Network-Wide Metro Ridership Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[18] Liu, Y., Z. Liu, and R. Jia, DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C: Emerging Technologies*, Vol. 101, 2019, pp. 18–34.

[19] Lin, L., Z. He, and S. Peeta, Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies*, Vol. 97, 2018, pp. 258–276.

[20] LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2324.

[21] Krizhevsky, A., I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] He, K., X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] Hu, J., L. Shen, and G. Sun, Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[24] Kingma, D. P. and J. Ba, Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.