# Multi-STGCnet: A Graph Convolution Based Spatial-Temporal Framework for Subway Passenger Flow Forecasting

Jiexia Ye, Juanjuan Zhao*, Kejiang Ye
*Shenzhen Institutes of Advanced Technology*
*Chinese Academy of Sciences*
Shenzhen, China
*University of Chinese Academy of Sciences*
Beijing, China
Email: {jx.ye,jj.zhao,kj.ye}@siat.ac.cn

Chengzhong Xu
*Faculty of Science and Technology*
*University of Macau*
Macau,China
Email: {czxu}@um.edu.mo

***Abstract*—Subway passenger flow forecasting, an essential component of intelligent transportation system, is critical for traffic management, public safety, urban planning. However, it is very challenging due to the high nonlinearities and complex dynamic spatio-temporal dependencies of passenger flows. In this paper, we model the subway system as a directed weighted graph and propose a novel spatio-temporal deep learning framework, Multi-STGCnet, for forecasting short-term subway passenger flow at a station level. Specifically, Multi-STGCnet is mainly composed of two components, temporal component and spatial component. (1) The temporal component employs three long short-term memory network (LSTM)-based modules to capture three temporal properties of the target station, which are the interval closeness, daily periodicity, weekly trend. (2) The spatial component designs three spatial matrixes to extract spatial correlation of a target station with all other stations classified as near neighbors, middle neighbors and distant neighbors. Respectively, it adopts three graph convolution network (GCN) and LSTM combined modules to capture the spatio-temporal influences from different neighbors. Finally, the outputs of the two components are fused with different weights to generate prediction. We evaluate Multi-STGCnet on a real world dataset from the metro system in Shenzhen, China. Experiment results demonstrate that our model outperforms multiple baselines.**

***Index Terms*—Passenger flow forecasting, GCN, LSTM, Spatial-Temporal Forecasting**

## I. Introduction

Over the last few decades, subway has experienced a rapid development around the world due to its capacity to solve traffic congestion. The automatic fare collection system (AFC) in subway system has generated very large amount of trip data and offers new opportunities for passenger flow prediction.

In this paper, we predict the short-term passenger flow of each subway station with historical observed data collected by AFC. However, it is very challenging due to the complex spatio-temporal correlations between the passenger flows of different stations during different time in whole metro system.

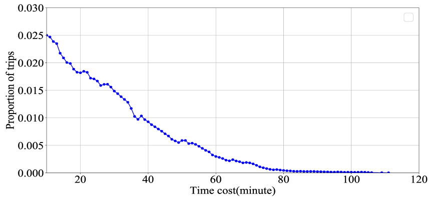


Fig. 1. The proportion of trips vs. time cost on historical AFC Data

(1) Spatial correlations. The inflow of the target station originates from the outflows of other stations and the mutual influence between any two stations is related to the distance between them. As shown in Fig. 1, the proportion of trips decreases as the time cost increases, indicating that passengers are more inclined to go to nearby stations than the far ones. This is consistent with the Tobler's first law of geography, which states that everything is related to everything else, but near things are more related to each other.

(2) Temporal correlations. From historical AFC data, we observe that the passenger flow of a station is related to its own historical observation, which can be summarized in three aspects: interval closeness, daily periodicity, weekly trend. Closeness refers that the previous flow will affect the current flow because the formation and dispersion of passengers are gradual. Daily periodicity indicates that the passenger flow is similar on consecutive days. Weekly trend indicates that there is a long term trend of the flows at the same time interval on the same day of weeks in a year.

Over the past few decades, there are many studies in traffic flow prediction. Time series methods such as Autoregressive Integrated Moving Average model (ARIMA) [1] are limited because they assume linear relationships among time lagged variables. Some tradition machine learning methods such as random forests, Support Vector Machine (SVM) [2], K-Nearest Neighbors (KNN) are able to model high-linearity

*Corresponding author: Juanjuan Zhao

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

in the traffic flow. However, they are sensitive to feature engineering and can't capture spatio-temporal dependencies. In recent years, frameworks based on deep learning have shown promising results for traffic flow prediction. The convolutional neural network (CNN) is usually employed to model the spatial properties of the traffic network through grid-based map segmentation. However, the subway system is a graph based structure in nature. Recently graph convolution networks (GCNs) have been developed to model such structured datasets [3]. However, they usually employ Laplacian matrix in GCN to extract the spatial dependency and it can't distinguish spatial properties of different regions with different distance.

In order to tackle these challenges, we model the subway network as a directed weighted graph and propose a novel deep learning framework Multi-STGCnet for predicted the outflow of the target station. Our main contributions are summarized as follows.

- Multi-STGCnet mainly consists of two components, a temporal component designed for modeling the temporal dependencies of passenger flows of the target station needed to predict, and a spatial component designed to capture the spatio-temporal impacts of all other stations on the target station.
- The temporal component summarizes the temporal dependencies related with the target station into the interval closeness, daily periodicity and weekly trend. It employs three LSTM based modules to model these temporal properties.
- The spatial component divides all other stations into near neighbors, middle neighbors and distant neighbors to distinguish different influence degrees on the target station. It adopts three GCN and LSTM combined modules, first extracting the spatial correlation with each kind of neighbors and then capturing the temporal dependencies based on historical observations of these neighbors.
- We novelly define a spatial matrix to replace the Laplacian matrix in traditional GCN. The spatial matrix is designed to represent the spatial structure of each kind of neighbors.
- We evaluate our approach using AFC data of Shenzhen metro system. The results demonstrate the advantages of our approach compared with other baselines. The codes of Multi-STGCnet are publicly available from https://github.com/start2020/Multi-STGCnet.

## II. Related Work

There are many different tasks on the domain of traffic prediction, such as subway passenger flow forecasting [4], taxi demand prediction [5] and citywide crowd flow prediction [6]. Though the datasets used by these tasks are variant, they all aim to predict the future traffic condition based on historical observation. Related works on traffic prediction provide references for our paper.

Time series methods such as HA, ARIMA [7], VAR [1] perform poorly on modeling high non-linearity problems. Traditional machine learning [8] including maximum likelihood estimation [4], KNN [9], SVM [10] are sensitive to feature engineering and can't achieve high accurate prediction. Deep learning with superior capacity to learn high non-linear properties without much domain knowledge has inspired researchers in transportation domain. Convolutional neural network (CNN) [6] [5] is widely used to capture the spatial correlation in traffic data by partitioning traffic network into a grid map. However, CNN is designed for grid based structure, which is not suitable for graph based structure, such as subway networks and highway networks, and it can not extract spatial dependencies of these networks accurately.

The recently developed graph convolutional network (GCN) is successfully adopted to generalize the tradition convolution to graph-structured data. Bruna *et al.* in 2014 [11] defined the filter of a graph in the Fourier domain, connecting spectral graph theory to deep learning. Defferrard *et al.* [12] in 2016 proposed fast localized convolutional filters on graphs to improve computational efficiency. The pilot work of Kipf and Welling [13] made a remarkable success in semi-supervised classification by using GCN. Yu *et al.* [3] proposed STGCN for traffic prediction on sensor network. One of the limitation of these works is that their models are constructed on undirected graph. There are some other works based on directed graph. Li *et al.* [14] proposed DCRNN on road network. Guo *et al.* [15] proposed ASTGCN to solve the highway flow forecasting. They can extract the temporal and spatial correlations on a road network. However, we argue that they overlook the different influence degree between regions with different distance when capturing spatial dependency.

Our proposed model is different from existing methods due to our problem setting. We investigate the subway traffic flow pattern and identify its unique spatiotemporal correlations. We model the subway network as a directed and weighted graph and define a spatial matrix based on the shortest path between stations to replace the Laplacian matrix of existing versions of GCNs, furthermore differentiating the mutual influence degrees between stations according to their distance.

## III. Preliminaries

In this section, we briefly define a subway network and formalize the subway passenger flow forecasting problem.

### A. Subway Network

A subway network is a directed graph $G = (V, E, A)$, where $V = \{v_1, v_2, \cdots, v_N\}$ is a set of nodes representing all stations in a subway, and $E$ is a set of directed edges referring the connectivity between two stations. If two nodes are two adjacent stations of a metro line, there is an edge between them. $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of graph G. The weight between two nodes represents the time taken by passengers to travel between the two stations. The weight of the edge between node $i$ and node $j$ is $w_{ij}$, representing the time needed to take traveling from station $i$ to station $j$.

### B. Formulation of Subway Passenger Flow Forecasting

We can observe two traffic features (inflow and outflow) of each node on the network at each time slice. Let us denote

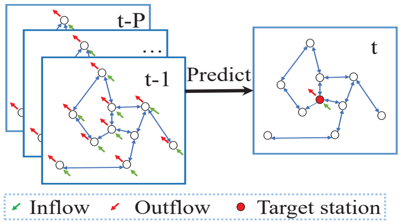


Fig. 2. Formulation of Subway Passenger Flow Prediction

the two features of node $i$ during $t$ time interval as $x_t^i = \{x_t^{i,1}, x_t^{i,2}\}, x_t^i \in \mathbb{R}^2$. During $t$ time interval, all features of the whole network can be denoted as a tensor $\boldsymbol{X}_t \in \mathbb{R}^{N \times 2}$.

Problem formulation: Given historical observations of the whole network over past $P$ time slices $\boldsymbol{X}_{t-P}, \cdots, \boldsymbol{X}_{t-2}, \boldsymbol{X}_{t-1}$, we predict the subway passenger flow of any target station in the next time slice $t$ (as shown in Fig.2).

$$[\boldsymbol{X}_{t-P}, \cdots, \boldsymbol{X}_{t-1}] \longrightarrow \boldsymbol{x}_t \tag{1}$$

## IV. Solution

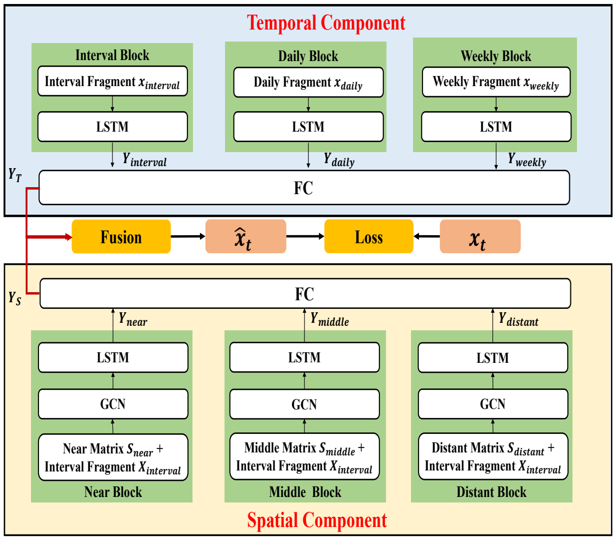

Temporal Component
Interval Block
Interval Fragment $x_{interval}$
LSTM
$Y_{interval}$
Daily Block
Daily Fragment $x_{daily}$
LSTM
$Y_{daily}$
Weekly Block
Weekly Fragment $x_{weekly}$
LSTM
$Y_{weekly}$
$Y_T$
FC
Fusion
$\hat{x}_t$
Loss
$x_t$
$Y_S$
FC
$Y_{near}$
$Y_{middle}$
$Y_{distant}$
LSTM
GCN
Near Matrix $S_{near}$ + Interval Fragment $X_{interval}$
Near Block
Middle Matrix $S_{middle}$ + Interval Fragment $X_{interval}$
Middle Block
Distant Matrix $S_{distant}$ + Interval Fragment $X_{interval}$
Distant Block
Spatial Component

Fig. 3. The architecture of Multi-STGCnet. GCN: Graph Convolution Network; LSTM: long short-term memory network; FC: Fully connected layer.

Multi-STGCnet is composed of two components: temporal component and spatial component as shown in Fig.3. The former is designed to extract temporal dependencies from the historical observation of the target station need to predict such as interval closeness, daily periodicity, weekly trend. The latter aims to model spatial influences from the near, middle, and distant neighbors of the target station.

Temporal component consists of three temporal blocks with the same structures. Each block is fed with the observed traffic flows of the target station on different time fragments and utilizes a long short-term memory network to capture the temporal correlation. The outputs of three blocks are merged by one fully connected layer. Spatial component is composed of three spatial-temporal blocks, which share the same structure but are fed with different spatial matrixes and traffic flows of different types of neighbors. Each block is composed of a graph convolution network modeling the spatial dependency and a long short-term memory network capturing the influence of historical traffic flows of each kind of neighbors on the target station. A fully connected layer is stacked to merge the outputs of three blocks. In the end, the outputs of the two components are further fused based on a parameter matrix to obtain the final result. The spatial matrixes and time fragments fed into Multi-STGCnet are defined as follows.

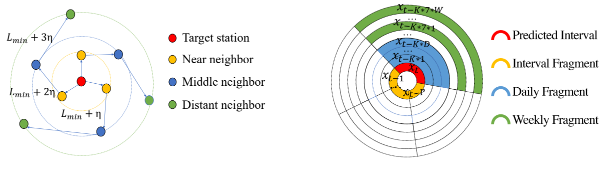
*1) Spatial Matrixes:* Intuitively, a nearby station is likely to have a larger impact than a distant station on the predicted target station. For simplicity, we divide all other stations into three kinds of neighbors according to their distances to the target station, which are near neighbors, middle neighbors and distant neighbors. We assume that different kinds of neighbors have different influence degrees on the target station and stations belonging to the same kind neighbor share similar impact. In addition, we define the distance between any two stations as the length of their shortest path for that most metro passengers tend to choose the path with distance as short as possible to save time.

Dijkstra algorithm is employed to find the shortest path with minimum time between each of other stations and the target station on the directed graph $G$. The set of length of all the shortest paths is defined as $L = \{l_j | j \in (1, ..., n)\}$, where $n$ is the number of stations and $l_j$ is the time cost of a shortest path between station j and the target station. The minimum and maximum value in the set is denoted as $L_{min} = argmin(L)$ and $L_{max} = argmax(L)$. We cut the difference between the maximum and minimum value into three equal parts as shown in Fig.4 (a), each part with the length $\eta = (L_{min} - L_{max})/3$. A station with $l \in (L_{min}, L_{min} + \eta)$ is regard as a near neighbor of the target station $O$. We use a $n$ dimension vector $V_O = [1, 0, 1, ..., 1]_{n \times 1}$ to represent all the near neighbors of station $O$ , where $i$-th element $v_{Oi} = 1$ refers that station $i$ is a near neighbor of station $O$ and $v_{Oi} = 0$ refers there is no near neighbor relationship between the two stations. Since each station has its own vector, we stack them together to define a Near Matrix, which contains the near neighbors information of the whole network. The Near matrix is denoted as follow:

$$S_{near} = [V_1, V_2, ..., V_n] = \begin{bmatrix} v_{11} & \ldots & v_{1n} \\ \ldots & \ldots & \ldots \\ v_{n1} & \ldots & v_{nn} \end{bmatrix} \tag{2}$$

where $v_{ij} = 0$ or $v_{ij} = 1$.

A station with $l \in (L_{min} + \eta, L_{min} + 2\eta]$ is regard as a middle neighbor of the target station $O$. Similarly, we define the Middle Matrix $S_{middle}$ which contains all the middle neighbors information of the whole network. A station with $l \in (L_{min} + 2\eta, L_{min} + 3\eta]$ is regard as a distant neighbor of the target station $O$ and we can denote the Distant Matrix $S_{distant}$ representing the distant neighbors information of the whole network. These three spatial matrixes can extract spatial properties of the subway network in different spatial aspects.



(a) Spatial Neighbors

(b) Time Fragments

Fig. 4. Spatial Temporal Features

*2) Time Fragments:* We divide one day into $K$ time intervals. Given a station, its passenger flow at the predicted time interval $t$ of a day, denoted as $x_t$, is correlated with its historical observations, which can be divided into three categories as shown in Fig.4 (b). (1) The interval fragment, refers to the flows at past $P$ time slices, defined as $x_{interval} = \left[\boldsymbol{x}_{(t-P)}, \cdots, \boldsymbol{x}_{(t-1)}\right]$. It represents the interval closeness property in traffic data. For instance, the traffic flow in 9am is inevitably influenced by previous intervals. Accordingly, we define the interval fragment of the whole network as $X_{interval} = \left[\boldsymbol{X}_{(t-P)}, \cdots, \boldsymbol{X}_{(t-1)}\right]$. (2) The daily fragment is the passenger flows at the same time interval as the predicted one on the past $D$ days, denoted as $x_{daily} = \left[\boldsymbol{x}_{(t-D*K)}, \cdots, \boldsymbol{x}_{(t-K)}\right]$. It represents the daily periodicity of traffic flows. For example, the peak hours are similar for workdays. (3) The weekly fragment refers to the flows at same time interval as the forecasting period in $W$ historical days with same week attributes, denoted as $x_{weekly} = \left[\boldsymbol{x}_{(t-K*7*W)}, \cdots, \boldsymbol{x}_{(t-K*7)}\right]$. It represents the weekly trend property in traffic data. Intuitively, there is a long-term trend of passenger flow in a year.

### A. Structure of Spatial Component

*1) GCN for Spatial Dependency Modeling:* Graph convolution network (GCN) is one of the state-of-art techniques for handling graph-based structure data, which generalizes classical convolutional neural network to the graph domain. GCN is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all vertices and $|\mathcal{V}| = n$ and $\mathcal{E}$ is the edge set. The graph structure is represented by a adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, representing the connections between vertices. The degree matrix is denoted as $D = \operatorname{diag}(d_1, d_2, \ldots, d_n) \in \mathbb{R}^{n \times n}$, where $d_i = \sum_j a_{ij}$ reflects the number of each node's neighbors.

In the spectral analysis, the properties of a graph can be represented by a Laplacian matrix, defined as $L = D - A \in \mathbb{R}^{n \times n}$. It has two versions of normalized form, which are defined as $L_{\text{sym}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ and $L_{\text{rw}} := D^{-1} L$ respectively. As Laplacian matrix is symmetric and semidefinite, it has a complete set of orthonormal eigenvectors $\{u_l\}_{l=0}^{n-1} \in \mathbb{R}^n$ with its associated set of eigenvalues, $\{\lambda_l\}_{l=0}^{n-1} \in \mathbb{R}^n$. Thus, its eigenvalue decomposition is that $L = U \Lambda U^T$ where $\Lambda = \operatorname{diag}([\lambda_0, \ldots, \lambda_{n-1}]) \in \mathbb{R}^{n \times n}, U = [u_0, \ldots, u_{n-1}] \in \mathbb{R}^{n \times n}$. As we can not apply the traditional convolution operator on a graph in vertex domain, the spectral theory defines the graph convolution operator in the Fourier domain [16]. The graph Fourier transform (GFT) of a spatial signal $x \in \mathbb{R}^n$ is defined as $\hat{x} = U^T x \in \mathbb{R}^n$, and its inverse as $x = U\hat{x}$ . The convolution operator on graph $*_{\mathcal{G}}$ is defined $x *_{\mathcal{G}} y = U\left(\left(U^T x\right) \odot \left(U^T y\right)\right)$, where $\odot$ is the Hadamard product. It follows that a graph convolution operation can be defined by:

$$y = \sigma\left(g_\theta *_{\mathcal{G}} x\right) = \sigma\left(U\left(\left(U^T g_\theta\right) \odot \left(U^T x\right)\right)\right) \tag{3}$$

$$y = \sigma\left(U g_\theta(\Lambda) U^T x\right) \tag{4}$$

where $g_\theta(\Lambda) = \operatorname{diag}(\theta)$ and $\theta \in \mathbb{R}^n$ is a vector of Fourier coefficients. However, it has to compute the eigenvalue decomposition of Laplacian matrix which is intolerable for the large scale graph. Another limitation is that it considers all the nodes in the graph during convolution and can't extract spatial localization. The limitation can be overcome by the replacement $g_\theta(\Lambda) = \sum_{k=1}^{K} \theta_k \Lambda^k$. The transform of the convolution is as follow:

$$y = \sigma\left(U \sum_{k=1}^{K} \theta_k \Lambda^k U^T x\right) = \sigma\left(\sum_{k=1}^{K} \theta_k L^k x\right) \tag{5}$$

The $K^{\text{th}}$ order polynomials of the Laplacian are K-localized [12]. Consequently, it is able to extract the information of 1 to $K^{\text{th}}$ order neighbors surrounding each node. Kpif and Welling in 2017 limited K=1 to produce the simplest graph convolution operation $y = \sigma(\theta L x)$. To rewrite it as a convolution layer, we can have

$$\boldsymbol{X}^{\boldsymbol{l}+\boldsymbol{1}} = \sigma\left(\boldsymbol{L}\boldsymbol{X}^{l} W\right) \tag{6}$$

where $\boldsymbol{X}^l$ denotes the $l$-th layer, $L$ is the Laplacian matrix, $W$ is the trainable parameters, $\sigma$ is the activation function, e.g, the sigmoid function or the ReLU function.

The subway network proposed in our paper is essentially a non-Euclidean graph structure. Therefore, we use GCN to model the spatial properties in the subway network. For scaling down the parameter spaces of our model, we divide all other stations into three categories: near, middle and distant neighbors. We want to aggregate the influence of stations in the same neighbors set and assign different weights to different neighbors set.

The matrix in GCN decides the scope of information that it can aggregate directly. However, the $1^{\text{th}}$ Laplacian matrix proposed by Kpif focuses on aggregating the information from the adjacent nodes. The$K^{\text{th}}$ power of Laplacian matrix proposed by Defferrard et al. is based on connectivity instead of the shortest path [12]. Both of them are unsuitable for our task. In this paper, we need a matrix which can aggregate the neighbors information with different distance based on shortest path theory.

We novelly define some spatial matrixs based on the shortest path algorithm (see its definition above) to replace the Laplacian matrix in GCN. The Near matrix is defined to represent the spatial structure of near neighbors of the target station. GCN with the Near matrix focuses on aggregating the information from the near neighbors directly. Likewise, GCN with the Middle matrix and Distant matrix focus on aggregating the information from the middle neighbors and distant neighbors respectively. We rewrite the GCN version proposed by Kipf and Welling as follow:

$$\boldsymbol{X}_{\boldsymbol{n}\times\boldsymbol{m}}^{l+1} = ReLU\left(\boldsymbol{S}_{n\times n}\boldsymbol{X}_{n\times c}^{l}W_{c\times m}\right) \tag{7}$$

where$\boldsymbol{X}_{n\times c}^{l}$ denotes the $l$-th layer with c features, $\boldsymbol{X}_{n\times m}^{l+1}$ denotes the next layer with m features, $\boldsymbol{S}_{n\times n}$ can be the near matrix $S_{near}$, the middle matrix $S_{middle}$ or the distant matrix $S_{distant}$ , $W_{c\times m}$ is the trainable parameters, ReLU is the activation function.

The input of each block of the spatial component is a spatial matrix $\boldsymbol{S}$ and an interval fragment $\boldsymbol{X}_{interval}$. The output of GCN in each block represents the aggregated information of each kind of neighbors.

*2) LSTM for Temporal Dependency Modeling:* The graph convolution operations extract spatial dependencies of the subway network in three different angles at each time slice, the output is further fed into LSTM to merge the information at the neighboring time slices of each kind of neighbors.

Long short-term memory network is a special type of Recurrent Neural Network(RNN), initially introduced by Hochreiter and Schmidhuber in 1997. It can overcome the exploding or vanishing gradient problem of RNN and exhibits the superior capability for time series prediction with long temporal dependency [17]. LSTM is composed of one input layer, one recurrent hidden layer and one output layer. The core of the hidden layer is a memory cell. We denote the state of the memory cell at time interval $t$ as $Carry_t$, which carries an accumulation of previous sequential information. In addition, there are three gates in the memory cell, namely input gate, forget gate and output gate. The input gate denoted as $In_t$ is used to input information at time interval $t$ to the network. The forget gate denoted as $Forget_t$ can forget some irrelevant information from the previous cell state $Carry_{t-1}$, while the output gate denoted as $Out_t$ controls the output of the memory cell. The structure of the memory cell in LSTM can be summarized as Euqation8, Where $\otimes$ denotes the Hadamard Product; $[W_i, U_i, b_i], i \in \{forget, in, out, new, y\}$ are all learnable parameters; $X_t$ is the features collected at time t. $Y_t$ is the output at time t.

$$\begin{aligned}
&Forget_t = \sigma(W_{forget}Hidden_{t-1} + U_{forget}X_t + b_{forget})\\
&In_t = \sigma\left(W_{in}Hidden_{t-1} + U_{in}X_t + b_{in}\right)\\
&Out_t = \sigma\left(W_{out}Hidden_{t-1} + U_{out}X_t + b_{out}\right)\\
&New_t = \sigma\left(W_{new}Hidden_{t-1} + U_{new}X_t + b_{new}\right)\\
&Carry_t = Forget_t \otimes Carry_{t-1} + In_t \otimes New_t\\
&Hidden_t = Out_t \otimes \tanh\left(Carry_t\right)\\
&Y_t = \mathrm{ReLu}\left(W_y Hidden_t + b_y\right)
\end{aligned} \tag{8}$$

LSTM's main objective is to model sequential dependencies and process arbitrary time lags for time series. These features are especially desirable for traffic prediction in the transportation domain. It can capture the influence of historical observation of different neighbors.

In conclusion, the spatial component is made up of three blocks to model the spatiotemporal impact from near, middle, distant neighbors. Each block first captures the spatial features and afterwards merges the influence of the historical observation from three kinds of neighbors in the subway network. The outputs of the near, middle, distant blocks are $Y_{near}$, $Y_{middle}$, $Y_{distant}$ respectively. Finally, a fully connected layer with an activation function as ReLu is employed to merge the information of the three blocks, generating the output denoted as $Y_S$.

### *B. Structure of Temporal Component*

The temporal component aims to model the temporal dependencies of a target station from its own historical observation. It is composed of three temporal blocks in charge of capturing the impact of interval closeness, daily periodicity and weekly trend. Every temporal block employs LSTM to predict the passenger flow during the next time interval based on the previous passenger flow.

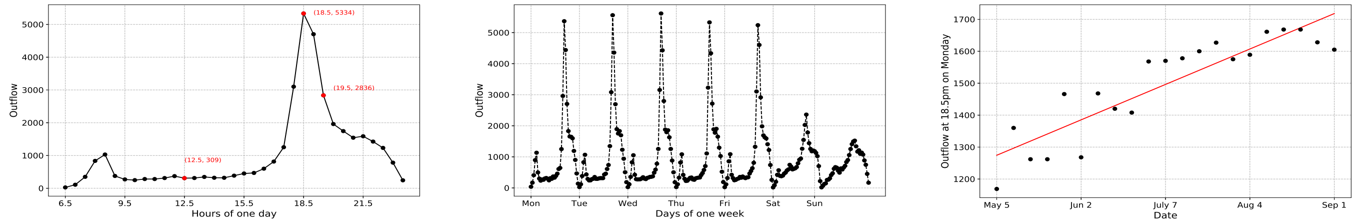
In a subway network, three temporal patterns can be observed explicitly in the passenger flow. (1) The interval closeness pattern refers that the current passenger flow is influenced largely by the passenger flow of its adjacent previous intervals instead of a distant interval. As shown in Fig. 5, the passenger flow at 19.5pm is more relevant to that at 18.5pm, rather than that at 12.5pm. The interval block is designed to model the interval closeness impact with interval fragment $x_{interval} = \left[\boldsymbol{x}_{(t-P)}, \cdots, \boldsymbol{x}_{(t-1)}\right]$ as input. (2) The daily periodicity impact implies that the passenger flows of the same time period in the past few days and the predicted period are similarly. The daily block captures the daily periodicity impact and its input is the daily fragment $x_{daily} = \left[\boldsymbol{x}_{(t-D*K)}, \cdots, \boldsymbol{x}_{(t-K)}\right]$. (3) The weekly trend impact describes the flow pattern of the same interval of the same working day or weekend from previous weeks. As shown in Fig. 5, there is an increasing trend of the passenger flow at 18.5pm on all Mondays from March to October. The trend block can model such impact, of which the input is the weekly fragment $x_{weekly} = \left[\boldsymbol{x}_{(t-W*K*7)}, \cdots, \boldsymbol{x}_{(t-K*7)}\right]$.

The output of the three temporal blocks are $Y_{interval}$, $Y_{daily}$, $Y_{weekly}$ respectively. A fully connected network is appended to merge the three outputs. The output of this component is $Y_T$.

### *C. Fusion*

This section aims to discuss how to integrate the outputs of the spatial component and the temporal component. If a target station is a transfer station, its passenger flow is largely depended on other stations, especially those directly connected with it. However, if a station is located at residential areas, it can expected that the majority of passengers enter or exit this station are those who live nearby. Thus its passenger flow is influenced more largely by its historical observation than that of other stations. Consequently, the influence degree of the two components are different for each station.

$$\hat{\mathbf{X_t}} = \mathbf{W}_S \odot \mathbf{Y}_S + \mathbf{W}_T \odot \mathbf{Y}_T \tag{9}$$



(a) Interval Closeness

(b) Daily Periodicity

(c) Weekly Trend

Fig. 5. Passenger Flow Patterns

where $\odot$ is the Hadamard product. $W_S$, $W_T$ are learning parameters, $Y_S$ is the output of the spatial component, $Y_T$ is the output of the temporal component, $\hat{\mathbf{X}}_{\mathbf{t}}$ is the prediction.

## V. Experiments

We carry out experiments on one real-world subway dataset from Shenzhen, one of first-tier cities in China, and evaluate performances of our model and other approaches.

### A. Datasets

The dataset used is collected by automatic fare collection system(AFC). The subway data is aggregated into every half an hour from the raw data for the reason that a horizon of 30 min is widely used in the analysis of transportation operations. The subway system in Shenzhen has 117 stations. The metro lines information and map are from Shenzhen Metro Group. The traffic measurements considered in our experiments are passengers inflow and passengers outflow. The time span of this dataset is from January to December in 2014, which is totally 12 months. We divide the first two weeks of each month into training set, the third week into validation set and the rest into test set, which guarantees an unbiased result.

### B. Baselines

We compare our model with the following methods.

*HA*: Historical average predicts the air quality of a station at a time interval by averaging the passenger flow of previous intervals for prediction.

*ARIMA*: Autoregressive integrated moving average is a well-known model for predicting time series data(Williams et al. 2003) [7] , it predicts the future passenger flow based on the previous passenger flow information based on the passenger flows of the station.

*LR*: Linear Regression is a simple model to extract linear correlation between variables.

*GBDT*: Gradient Boosting Decision Tree (Friedman et al. 2001) [18] is an ensemble learning method composing of many regression decision trees sequentially.

*XGBoost*: XGBoost (Chen et al. 2016) [19] is conceptually similar to GBDT but also differs in many aspects. It can handle with linear and non-linear features and leads to a very good generalization.

*RF*: Random Forest (Breiman et al. 2001) [20] is a combination of many regression tree predictors whose prediction is the average prediction of all the trees.

*ANN*: The Artificial Neural Network in our paper is a three fully connected layers with 32, 12, 1 units respectively.

*GCN*: Graph convolution network (Defferrard et al. 2016) [12] successfully generalizes CNN to graph-structured data.

*GRU*: Gated Recurrent Unit (Chung et al. 2014) [21] is a simplified version of LSTM but is less likely to overfit.

### C. Loss Function and Evaluation Metrics

We choose MAE (mean absolute error) as the loss function. MAE, RMSE (root mean square error) and MAPE (mean absolute percentage error) are the metrics to asses model performances. Their definitions are as follows, where $y$ is the ground truth, $\tilde{y}$ is the prediction.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\tilde{y}_i - y_i|}{y_i} * 100\% \tag{10}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\tilde{y} - y_i| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tilde{y} - y_i)^2} \tag{12}$$

Prediction is made on station level and the metrics are computed as the average performance of all the stations.

### D. Experiment Results

TABLE I
Average Performance Comparison with Various Approaches

| Model | MAE | MAPE | RMSE |
|---|---|---|---|
| HA | 387.523174 | 0.843366 | 740.087909 |
| ARIMA | 224.554015 | 0.501994 | 395.194652 |
| LR | 218.807822 | 0.403183 | 389.382387 |
| XGB | 74.084914 | 0.157338 | 117.887879 |
| GBDT | 60.523768 | 0.104170 | 104.813541 |
| RF | 56.736383 | 0.089223 | 99.209742 |
| ANN | 68.836687 | 0.134655 | 117.765380 |
| GCN | 63.749587 | 0.131476 | 112.284874 |
| GRU | 50.303431 | 0.095632 | 84.426589 |
| **Multi-STGCnet** | **46.344920** | **0.072743** | **69.836269** |

We compare Multi-STGCnet with various methods in terms of metrics MAPE, MAE, RMSE. The best performance are highlighted with bold font. As can be seen from Table I, our proposed Multi-STGCnet has the lowest MAE (46.34), the lowest MAPE (0.072) and the lowest RMSE (69.83) among all the methods. More specifically, it can be observed that traditional statistical methods such as HA, ARIMA, LR have the worst performances, as they lack the capacity to extract complex non-linearity in subway data. On the other hand, ensemble methods of machine learning including XGB, GBDT, RF achieve much better performances, proving that they can capture the non-linear correlation between the current passenger flow and its historical observation. Among the deep learning method, GRU achieves a lower MAE (50.30), a lower RMSE (84.42) and a lower MAPE (0.095) than the other two deep learning methods ANN and GCN. ANN can not extract the sequential pattern of time series data while GRU has the superior capability for time series prediction with long temporal dependency. This demonstrates that there is a long temporal dependency in subway data. GCN focuses on modeling spatial dependencies of the subway network. The fact that its performance is worse than GRU largely indicates that the dependency in temporal dimension is stronger than the dependency in spatial dimension.

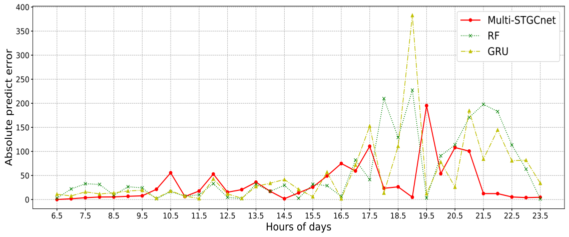


Fig. 6. Comparison of absolute predict error among top three approaches for one day

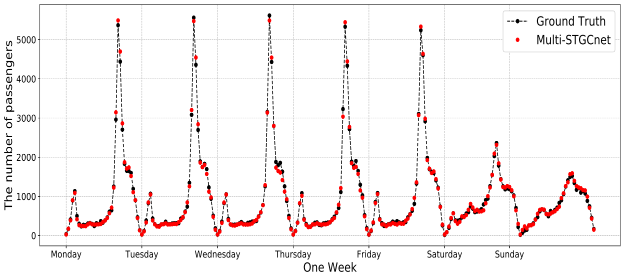


Fig. 7. Prediction of passenger flow for one week by Multi-STGCnet

Compared with GRU modeling only temporal correlation from historical observation of the target station and GCN only modeling spatial correlation from surrounding stations, our model has the capability of capturing both spatial and temporal correlation of the whole network. We compare our model with the best machine learning method RF and the best deep learning model GRU in our paper by absolute predict error, as shown in Fig. 6. We find that both RF and GRU behave worse in the evening, specifically, from 17.5pm to 23pm while STSP-MGCN has an ideal prediction of these hours. Finally, Multi-STGCnet not only significantly outperforms all the other methods but also achieves an accurate passenger flow prediction. For example, it can observed that the prediction of Multi-STGCnet is very close to the ground truth in one week on November(as shown in Fig.7).

### *E. Model Component Analysis*

Multi-STGCnet is composed of two independent components and each component consists three blocks. Every block is in charge of extracting different spatial temporal features from subway traffic data. In order to measure the contribution of each block for subway passenger flow prediction, we evaluate each component and each block independently.

(1) *Interval block*: This block is responsible to extract the temporal dependencies from historical flow of the target station during the past $P$ interval time.

(2) *Daily block*: This block aims to model the temporal correlation of passenger flow at the same intervals during the past $D$ days of the target station.

(3) *Weekly block*: This block is in charge of capturing the long term trend in passenger flow. We use the previous observation of the same period on the last $W$ weeks for prediction.

(4) *Multi-STGCnet-T*: The temporal component is a combination of interval block, daily block, weekly block. It extracts the dependencies from the historical observation of the target station in three temporal dimension aspects, including the interval closeness, daily periodicity and weekly trend.

(5) *Near block*: This block can extract spatial temporal dependencies from near neighbors of the target station.

(6) *Middle block*: This block is able to model the influence from middle neighbors in both spatial and temporal dimension.

(7) *Distant block*: This block models the impact of passenger flow from distant neighbors.

(8) *Multi-STGCnet-S*: The spatial component is designed to merge the spatiotemporal influence from the all other stations on the target station.

TABLE II
PERFORMANCE COMPARISON AMONG DIFFERENT BLOCKS OF MULTI-STGCNET

| Model | MAE | MAPE | RMSE |
|---|---|---|---|
| Interval block | 48.598836 | 0.078841 | 79.409736 |
| Daily block | 64.530737 | 0.124871 | 99.247472 |
| Weekly block | 51.352493 | 0.076651 | 83.758802 |
| Multi-STGCnet-T | 46.879129 | 0.084599 | 77.280309 |
| Near Block | 50.266971 | 0.085582 | 82.086010 |
| Middle Block | 55.571977 | 0.090005 | 91.010304 |
| Distant Block | 68.961511 | 0.087172 | 115.447569 |
| Multi-STGCnet-S | 50.049825 | 0.079537 | 79.859913 |
| **Multi-STGCnet** | **46.344920** | **0.072743** | **69.836269** |

Table II shows the performances of Multi-STGCnet and its components and blocks. On one hand, it can be observed that interval block achieves a lower MAPE and RMSE than daily block and weekly block. It demonstrates that the historical passenger flow of the adjacent previous intervals has a larger impact on the current flow than those on previous days. In addition, daily block performs worse than weekly block on every metric, proving that passenger flow pattern is more related with the historical days which have the same week attribute. For instance, the traffic pattern on Monday is usually more close to that on historical Mondays than that on other weekdays or weekends. Comparatively, the performance achieves the best when the three blocks are combined, which demonstrates the effectiveness of considering impact of interval closeness, daily periodicity and weekly trend together.

On the other hand, in terms of MAE and RMSE, near block outperforms middle block while middle block performs better than distant block. It indicates further that the mutual influence degree between two stations changes as their distance changes The results indicates further that the mutual influence degree between two stations is negative correlated with the distance between them. When the impact from all the rest stations is considered, the performance achieves best and it demonstrates the effectiveness of the spatial component. Finally, the combination of Multi-STGCnet-T and Multi-STGCnet-S has the best performance on all the metrics and it proves that our elaborate model has a superior capacity of modeling the dependencies in both spatial and temporal dimension.

## VI. Conclusion and Future work

In this paper, we study the passenger flow prediction at a station level on a graph-based subway network. We novelly propose a graph convolution based spatial-temporal deep learning framework, Multi-STGCnet. In our model, we adopt LSTM based modules to extract temporal correlation between historical observation of the target station and its current flow in three aspects, namely interval closeness, daily periodicity and weekly trend. We employ GCN-LSTM based modules to capture the spatiotemporal dependencies from all other stations in the subway network. What's more, we distinguish the mutual influence degree between the target station and all other stations by dividing them into near neighbors, middle neighbors and distant neighbors according to their shortest distance to the target station. In addition, we novelly define a spatial matrix to represent the spatial features of each kind of neighbors, replacing the Laplacian matrix in GCN. The evaluation on Shenzhen metro dataset demonstrates the superiority of Multi-STGCnet.

External factors such as special events, weather also have complex correlation with the subway passenger flow. In the future, we would like to consider these factors to improve the accuracy of prediction. Multi-STGCnet is a general model designed for spatiotemporal forecasting on a graph-based network, which is a reference for any application shares the same demand.

## VII. Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments.This work is supported in part by the National Key R&D Program of China (No. 2019YFB2102100), and by "National Natural Science Foundation of China" No. 61802387, and by Science and Technology Development Fund of Macao S.A.R (FDCT) under number 0015/2019/AKP, and by Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence, and by China's Post-doctoral Science Fund No.2019M663183, and by National Natural Science Foundation of Shenzhen No.JCYJ20190812153212464.

## References

[1] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-Plus®*, pp. 385–429, 2006.

[2] Y.-n. Yang and H.-p. Lu, "Short-term traffic flow combined forecasting model based on svm," in *2010 International Conference on Computational and Information Sciences*. IEEE, 2010, pp. 262–265.

[3] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, 2018, pp. 3634–3640.

[4] J. Zhao, F. Zhang, L. Tu, C. Xu, D. Shen, C. Tian, X.-Y. Li, and Z. Li, "Estimation of passenger route choice pattern using smart card data for complex metro systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, 2017.

[5] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[8] H. Chen, S. Wang, Z. Deng, X. Zhang, and Z. Li, "Fgst: Fine-grained spatial-temporal based regression for stationless bike traffic prediction," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2019, pp. 265–279.

[9] Y. Djenouri, A. Belhadi, J. C.-W. Lin, and A. Cano, "Adapted k-nearest neighbors for detecting anomalies on spatio–temporal traffic flow," *IEEE Access*, vol. 7, pp. 10 015–10 027, 2019.

[10] X. Ling, X. Feng, Z. Chen, Y. Xu, and H. Zheng, "Short-term traffic flow prediction with optimized multi-kernel support vector machine," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 294–300.

[11] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations, ICLR*, 2014.

[12] 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.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017, pp. 1–12.

[14] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *International Conference on Learning Representations, ICLR*, 2018.

[15] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 922–929.

[16] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[17] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[18] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[19] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.