STP-TrellisNets: Spatial-Temporal Parallel TrellisNets for Metro Station Passenger Flow Prediction

Junjie Ou[1], Jiahui Sun[1], Yichen Zhu[1], Haiming Jin[1*], Yijuan Liu[1], Fan Zhang[2], Jianqiang Huang[3], Xinbing Wang[1*]
[1]Shanghai Jiao Tong University, [2]SIAT, Chinese Academy of Sciences, [3]Alibaba Damo Academy
{j_michael,jhsun1997,zyc_ieee,jinhaiming,yooeachen,xwang8}@sjtu.edu.cn,zhangfan@sibat.cn,jianqiang.jqh@gmail.com

ABSTRACT

Recent years have witnessed a drastic increase in the number of urban metro passengers, which inevitably causes the overcrowdedness in the metro systems of many cities. Clearly, an accurate prediction of passenger flows at metro stations is critical for a variety of metro system management operations, such as line scheduling and staff preallocation, that help alleviate such overcrowdedness. Thus, in this paper, we aim to address the problem of *accurately predicting metro station passenger (MSP) flows*. Similar to other traffic data, such as road traffic volume and highway speed, MSP flows are also spatial-temporal in nature. However, existing methods for other traffic prediction tasks are usually suboptimal to predict MSP flows due to MSP flows' unique spatial-temporal characteristics. As a result, we propose a novel deep learning framework *STP-TrellisNets*, which for the first time augments the newly-emerged temporal convolutional framework *TrellisNet* for spatial-temporal prediction. The temporal module of STP-TrellisNets (named *CP-TrellisNets*) employs two TrellisNets in serial to jointly capture the short- and long-term temporal correlation of MSP flows. In parallel to CP-TrellisNets, its spatial module (named *GC-TrellisNet*) adopts a novel transfer flow-based metric to characterize the spatial correlation among MSP flows, and implements multiple *diffusion graph convolutional networks (DGCNs)* in time-series order with their outputs connected to a TrellisNet to capture the dynamics of such spatial correlation. Clearly, GC-TrellisNet essentially integrates TrellisNet with graph convolution, and empowers TrellisNet with the ability to capture dynamic graph-structured correlation. We conduct extensive experiments with two large-scale real-world automated fare collection datasets, which contain respectively about 1.5 billion records in Shenzhen, China and 70 million records in Hangzhou, China. The experimental results demonstrate that STP-TrellisNets outperforms the state-of-the-art baselines.

CCS CONCEPTS

• Information systems → Spatial-temporal systems; **Data mining**; **• Computing methodologies → Neural networks**.

*Corresponding authors.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411874

KEYWORDS

Metro Station Passenger Flow Prediction; TrellisNet; Diffusion Graph Convolution; Dynamic Spatial Correlation

ACM Reference Format:
Junjie Ou, Jiahui Sun, Yichen Zhu, Haiming Jin, Yijuan Liu, Fan Zhang, Jianqiang Huang, Xinbing Wang. 2020. STP-TrellisNets: Spatial-Temporal Parallel TrellisNets for Metro Station Passenger Flow Prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411874

1 INTRODUCTION

Nowadays, with the rapid development of metro systems, a vastly increasing number of urban residents choose to take the metro for their daily transportation (e.g., commuting, shopping). However, such a drastic increase in the number of metro passengers has apparently overcrowded the metro systems in many cities, which threatens the public safety and exacerbates the difficulty of metro system management. Clearly, knowing beforehand the number of passengers entering and exiting metro stations is critical for proper metro line scheduling and timely metro staff preallocation, which could help handle the aforementioned overcrowdedness in metro systems. Therefore, in this paper, we study the problem of *metro station passenger (MSP) flow prediction* with the data collected from the automated fare collection (AFC) system.

In practice, the MSP flows in a metro system usually show strong spatial-temporal correlations. That is, the MSP flow of one station is both temporally correlated with those in the past, and spatially correlated with those of other stations. As a result, capturing such spatial-temporal correlations is essential for accurately predicting MSP flows. Although a series of studies have effectively accomplished other spatial-temporal prediction tasks [1–10], such as traffic volume and highway speed prediction, they invariably become suboptimal for MSP flow prediction due to the unique spatial-temporal characteristics of MSP flows. In what follows, we will elaborate upon the design of our novel deep learning framework optimized to better capture MSP flows' spatial-temporal correlations.

From the **temporal** perspective, RNN-based models [8–12] have been regarded as effective to capture the non-linear temporal correlation of short time series. However, the MSP flows at least in the past few hours have to be considered to predict those in the future. Clearly, the excessive number of data samples contained in such long historical MSP flows will inevitably make the training of RNN rather inefficient, because of its inherent recurrent architecture and serial operations. Thus, instead of RNN, we adopt the *Temporal Convolutional Network (TCN)* [13, 14], whose parallel

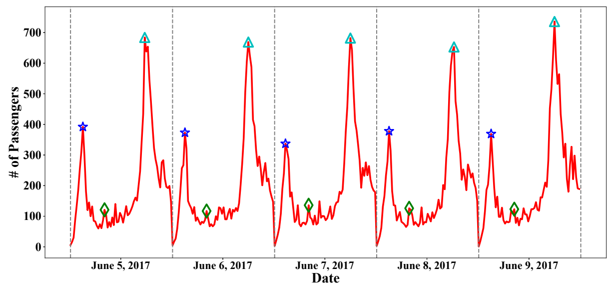
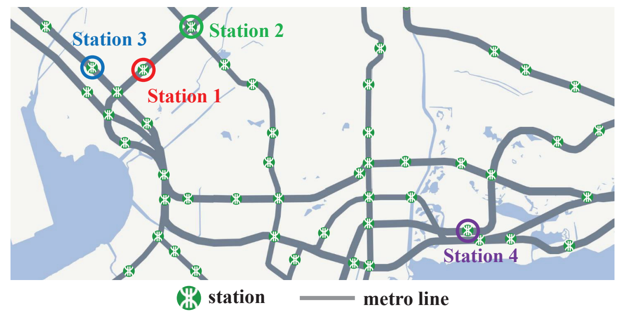
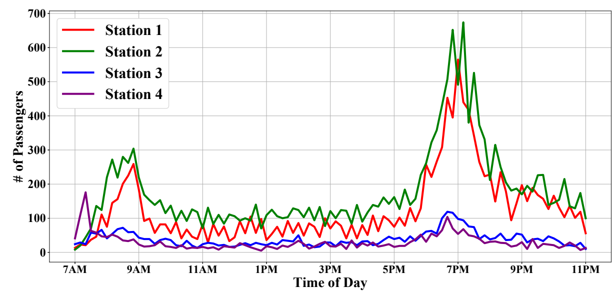


Figure 1: An example that shows the approximate periodicity of MSP flows, where the curve represents the MSP flows in 5 consecutive days of metro station Fanshen in Shenzhen and the same marker corresponds to the same time of day.



(a) The 4 stations selected in the Shenzhen metro system.



(b) The MSP flows of the 4 stations marked in Figure 2(a) in one day.

Figure 2: The MSP flows of 4 metro stations in Shenzhen in one day, where close stations show similar (e.g., Stations 1 and 2) or distinctive (e.g., Stations 1 and 3) flow patterns, and distant stations show similar (e.g., Stations 3 and 4) or distinctive (e.g., Stations 1 and 4) flow patterns as well.

framework combined with its exponentially enlarging receptive fields across the convolution layers enables much faster training than RNN. More specifically, we employ a novel TCN variant called *Trellis Network (TrellisNet)* [15], which inherits the above merits of the traditional TCN and additionally enhances its ability to capture the non-linearity of time series by combining input injection with a non-linear transformation.

Though promising, TrellisNet could not unleash its full power unless the temporal characteristics of MSP flows are appropriately taken into consideration. On one hand, as shown in Figure 1, the MSP flow of a station typically shows a similar pattern within each day. If handled appropriately, such approximate periodicity could offer useful side information for predicting future flows. On the other hand, the MSP flows between two adjacent days are naturally discontinuous, because the metro system typically closes between mid-night and early morning. Properly dealing with such flow discontinuity then becomes another critical factor that affects the prediction performance.

To address the above problems, we propose a novel framework named *Closeness-Periodicity TrellisNets (CP-TrellisNets)* with two TrellisNets in serial as its basic building blocks. Within such framework, a *Closeness TrellisNet (C-TrellisNet)* takes as input the MSP flows in the past few hours, whose output is then *skip-concatenated* with those at the same time instance in the past few days and is further fed to a *Periodicity TrellisNet (P-TrellisNet)*. In this way, the output of the P-TrellisNet integrates both the short-term and long-term temporal features of the MSP flow, with the skip-concatenation effectively addressing the flow discontinuity issue.

From the **spatial** perspective, using TrellisNets alone is usually far from enough to capture the spatial correlations of MSP flows. Inspired by the recent success of graph convolutional networks (GCNs) [9–14] on learning graph-structured correlations, we propose to empower TrellisNet with the ability to learn spatial correlations by integrating it with GCNs.

The first step towards such integration is to construct an appropriate graph representation for the metro system. As shown in Figure 2, geographical attributes, such as distance and connectivity, which oftentimes well characterize the spatial correlation in other scenarios, do not necessarily determine that among MSP flows. In practice, the MSP flow of one station depends on the passenger flows transferred between it and other stations, and thus the transfer flows could naturally serve as the metric to represent the spatial correlation in our metro scenario. As a result, we propose to represent the metro system as a *metro direct graph* with vertices corresponding to metro stations, and edge weights proportional to the *transfer flows* between pairs of stations.

Clearly, the edge weights of the metro direct graph are dynamic due to the time-varying nature of transfer flows. In order to capture such dynamic spatial correlation, we propose a novel framework named *Graph Convolutional TrellisNet (GC-TrellisNet)*, which integrates TrellisNet with graph convolutions. More specifically, GC-TrellisNet contains multiple *Diffusion Graph Convolutional Networks (DGCNs)* [16] placed along the time steps with each of them taking the metro directed graph at the corresponding time step as input. Then, the outputs of these DGCNs are combined and fed to a *Dynamic TrellisNet (D-TrellisNet)* in time-series order. In this way, the output of the D-TrellisNet captures the aforementioned dynamic spatial correlation.

Collectively from the **spatial-temporal** perspective, we aggregate the outputs of the parallel CP-TrellisNets and GC-TrellisNet to obtain the predicted MSP flows. As a result, the overall architecture named *Spatial-Temporal Parallel TrellisNets (STP-TrellisNets)* could well capture the short- and long-term temporal correlation, as well as the dynamic spatial correlations of MSP flows.

To summarize, this paper makes the following contributions.

- In this paper, we address the MSP flow prediction problem by proposing STP-TrellisNets, a novel deep learning framework which learns the spatial-temporal correlations of MSP flows. To the best of our knowledge, this paper is the first one that adopts and augments TrellisNets for spatial-temporal prediction tasks.

- We propose a novel framework CP-TrellisNets which implements two TrellisNets in serial with a skip-concatenation to jointly capture the short- and long-term temporal correlation of MSP flows and address the flow discontinuity issue.
- We design a novel transfer flow-based metric to represent the spatial correlations among MSP flows, and further propose a novel framework GC-TrellisNet consisting of multiple DGCNs in time-series order with their outputs connected to a TrellisNet which captures the dynamics of such correlation.
- We conduct extensive experiments on two real-world large-scale AFC datasets with about 1.5 billion records within 30 days in Shenzhen, China and about 70 million records within 25 days in Hangzhou, China. The experimental results show that our proposed STP-TrellisNets outperforms the existing baseline methods.

2 RELATED WORK

Traffic prediction plays a key role in traffic management, and has received wide attention over the years. Essentially, traffic prediction aims at predicting future values of traffic conditions, such as taxi demand [17], traffic speed [11] and crowd flows [18].

Inspired by the ability of RNN to model the correlation of sequential data, RNNs [8, 9] were adopted for traffic prediction problems. However, since RNN suffers from gradient vanishing, it is usually hard for it to handle long-range input sequences. While several variants of RNN have been proposed to address this problem (e.g., LSTM [10], GRU [11]), they still suffer from high memory requirement and time-consuming training due to its recurrent architecture. Instead of employing RNN, our proposed STP-TrellisNets effectively exploits the strengths of a novel TCN variant TrellisNet on handling the long-range input sequence and capturing the non-linearity of MSP flows. Specifically, our STP-TrellisNets has an architecture with two TrellisNets in serial to jointly take the long- and short-term temporal correlation of MSP flows into consideration.

Recently, several studies further considered spatial correlation by treating the traffic data as a regular image [17–21] or a graph [7, 9–14, 22–25] with geographical attributes (e.g., distance, connectivity) among city regions, road segments, etc., to represent the spatial correlation. Among them, [17] proposed a local CNN method on the image to consider the correlation of taxi demand among spatially nearby city regions. [18] employed a ConvPlus model on the grid structure of city to model the long-range spatial dependency among the crowd flows in different regions. [7] employed a spectral graph convolution on the graph to extract the meaningful spatial features of road traffic speed. In addition, with self-attention mechanism, [23, 24] could capture more global spatial correlations from the graph-structured data. These works either modeled spatial correlation in terms of geographical attributes or assumed static spatial correlation. However, in our metro scenario, spatial correlation is not consistent with geographical attributes and changes in a dynamic way, which make them not fully applicable. Although [26, 27] introduced some dynamics into the graph representation of GCN's input, such graph structure is essentially still based on geographical attributes, which is far from enough to capture the dynamic spatial correlation among MSP flows. Different from them, in this paper, we propose to represent the metro system as a metro directed graph based on a novel transfer flow-based metric, because the transfer flows among metro stations determine their MSP flows. Furthermore, we integrate the diffusion graph convolution with TrellisNet to capture the dynamics of MSP flows' spatial correlation.

Several recent works [28–35] specifically paid attention to the metro scenario. [28] addressed the crowd flow distribution prediction problem across the entire train network. [29, 30] estimated the route choices of passengers in complex metro networks. Clearly, the problems addressed by these three works are different from the station-level passenger flow prediction problem studied in this paper. [31, 32] employed 1D CNN to capture the spatial correlation of stations in one single metro line, which is not suitable for a practical metro system with multiple metro lines. [33] gave an empirical study to predict MSP flows, but the proposed model can hardly handle the nonlinearity and complexity of metro traffic data. [34] employed a sequence learning model to predict the passenger flows but neglected the long-term periodicity and spatial correlation of MSP flows. Furthermore, [35] modeled the MSP flows as the weakly-dependent tensor data and designed a tensor completion algorithm to address the prediction problem. However, such dependency was characterized by static geographical and contextual attributes, which is suboptimal for representing MSP flows' dynamic spatial correlation. Different from them, our proposed STP-TrellisNets focuses on addressing the MSP flow prediction problem in complex real-world metro systems with dedicatedly designed modules to capture both the long- and short-term temporal correlation and dynamic spatial correlation of MSP flows.

3 METHODOLOGY

In this section, we give a formal description of the MSP flow prediction problem, as well as an overview and the design details of the proposed STP-TrellisNets.

3.1 Overview

We consider an urban metro system with a set of n stations denoted as $\{1, 2, \cdots, n\}$ and split the overall timeline (e.g., 1 month) into a set of time intervals, denoted as $\{0, 1, \cdots, q\}$, with equal length (e.g., 30 minutes). Next, we introduce in Definition 1 the concepts of MSP inflow and outflow and in Definition 2 the MSP flow prediction problem addressed in this paper.

Definition 1 (MSP Inflow and Outflow). *For a metro station i, the MSP inflow $f_{i,t}^{\text{in}}$ and outflow $f_{i,t}^{\text{out}}$ in time interval t are defined respectively as the number of passengers entering and exiting station i during time interval t.*

Definition 2 (MSP Flow Prediction Problem). *Given the historical flow data of the metro system until time interval t, the MSP flow prediction problem aims to predict the MSP outflow of each metro station*[1] *in the next time interval $t + 1$.*

To address the above MSP flow prediction problem, we propose a novel deep learning framework *Spatial-Temporal Parallel TrellisNets (STP-TrellisNets)* shown in Figure 3, which consists of a temporal module *Closeness-Periodicity TrellisNets (CP-TrellisNets)*, a spatial

[1]In this paper we focus on predicting MSP outflow, and leave MSP inflow prediction in our future work.

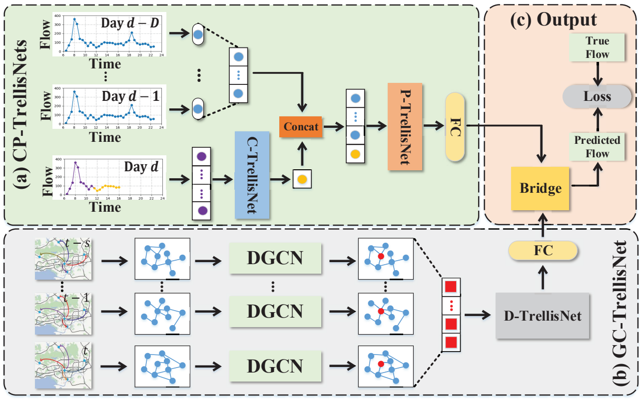


Figure 3: The overall architecture of STP-TrellisNets, where d represents the current day and FC denotes the Fully Connected layer.

module *Graph Convolutional TrellisNet (GC-TrellisNet)*, and an aggregation module Output. The CP-TrellisNets takes the historical MSP outflows as input, and outputs a result that captures both the short-term and long-term temporal correlation of MSP outflows. The GC-TrellisNet takes the historical MSP inflows and multiple metro directed graphs along the time steps as input, and outputs a result that captures the dynamic spatial correlation of MSP outflows. Then, the Output module employs a Bridge operation to aggregate the outputs from the CP-TrellisNets and the GC-TrellisNet into the final prediction result. The details of each part will be described in the following Sections 3.2, 3.3, and 3.4.

3.2 Long Short-Term Temporal Correlation: CP-TrellisNets

In this work, we adopt a special temporal convolutional network characterized by weight sharing and input injection with a non-linear transformation called *TrellisNet* [15] to capture the non-linear temporal correlation of MSP flows. The reasons for which we adopt TrellisNet are two-fold. On one hand, its temporal convolutional architecture with a weight sharing mechanism enables stable gradient, fast training speed, and exponentially large receptive field size to better handle the long historical MSP flow sequence than recurrent architectures. On the other hand, its input injection, which integrates the deep features with the original input sequence, followed by a non-linear transformation improves the performance of the temporal convolutional architecture for capturing the non-linearity of MSP flows.

Considering the discontinuity between the long-term and short-term historical MSP outflows, we propose the CP-TrellisNets which firstly uses a Closeness TrellisNet (C-TrellisNet) to capture the short-term temporal correlation, followed by another Periodicity-TrellisNet (P-TrellisNet) to further capture the long-term temporal correlation.

3.2.1 Short-Term Temporal Correlation: C-TrellisNet. For the whole metro system, in time interval t, we use an n-dimensional vector $\mathbf{x}_t = (f^{\text{out}}_{1,t}, f^{\text{out}}_{2,t}, \cdots, f^{\text{out}}_{n,t}) \in \mathbb{R}^n$ to represent the MSP outflows of the n stations. To predict the MSP outflows in time interval $t+1$, we use the MSP outflows in the past T time intervals as the historical input data. As a result, we have a length-T MSP outflow data sequence $\mathbf{X}_{1:T}$ as the input of C-TrellisNet given as

$$\mathbf{X}_{1:T} = \left(\mathbf{x}_{t-(T-1)}, \mathbf{x}_{t-(T-2)}, \cdots, \mathbf{x}_t\right).$$

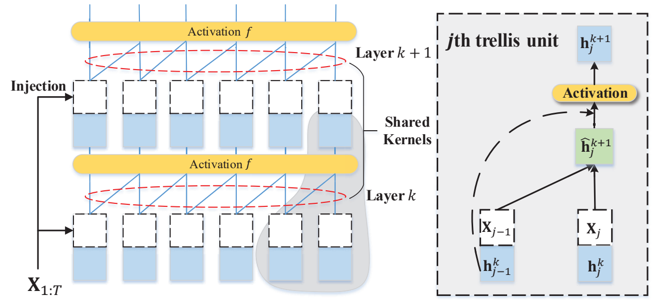


Figure 4: An example of the TrellisNet architecture for MSP flow prediction.

As illustrated in Figure 4, a full TrellisNet is built by stacking its basic units across the time steps and layers to constitute a trellis-like network. Specifically, the jth unit in layer k of a TrellisNet consists of the hidden output $\mathbf{h}^k_{j-1} \in \mathbb{R}^n$ and $\mathbf{h}^k_j \in \mathbb{R}^n$ from the previous layer $k-1$ and an injection of vectors $\mathbf{X}_{j-1}$ and $\mathbf{X}_j$ from the input sequence. The transformation in the jth unit is represented as the following Equation (1),

$$\begin{cases} \widehat{\mathbf{h}}^{k+1}_j = W_1\left[\mathbf{X}_{j-1} \parallel \mathbf{h}^k_{j-1}\right] + W_2\left[\mathbf{X}_j \parallel \mathbf{h}^k_j\right] \\ \mathbf{h}^{k+1}_j = f\left(\widehat{\mathbf{h}}^{k+1}_j, \mathbf{h}^k_{j-1}\right) \end{cases}, \tag{1}$$

where $\widehat{\mathbf{h}}^{k+1}_j \in \mathbb{R}^n$ is the pre-activation output, $\mathbf{X}_{j-1}$ and $\mathbf{X}_j$ correspond to the $(j-1)$th and jth vector in the input sequence, $||$ is the concatenation operator, W_1 and W_2 are kernel weights, $\mathbf{h}^{k+1}_j \in \mathbb{R}^n$ represents the output of the jth unit in layer k, and $f: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear activation function applied to $\widehat{\mathbf{h}}^{k+1}_j$ and $\mathbf{h}^k_{j-1}$.

We apply the above transformation procedure across all time steps and all layers as shown in Figure 4, using the same kernel weight matrix. Mathematically, given the input MSP outflow data sequence $\mathbf{X}_{1:T}$, the computation at each layer k of C-TrellisNet can be summarized as

$$\mathbf{h}^{k+1}_{1:T} = f\left(\left(\mathbf{h}^k_{1:T} \parallel \mathbf{X}_{1:T}\right) * \mathbf{W},\ \mathbf{h}^k_{1:T-1}\right), \tag{2}$$

where $*$ denotes the 1D causal convolution operation with zero padding that convolves the output of the previous layer with only data from the past time intervals, and $\mathbf{W}$ denotes the kernel weight matrix parameters which is shared across all layers. Note that, similar to the generic temporal convolution network, we add a dilation factor in the 1D convolution operation to enlarge our C-TrellisNet's receptive fields.

At level 0, we initialize $\mathbf{h}^0_{1:T} = \mathbf{0}$. After stacking s layers, we take the last output of the sth layer (i.e., $\mathbf{h}^{s+1}_T$) which is an n-dimensional vector as the output of C-TrellisNet, i.e.,

$$\Gamma^{(s)}(\mathbf{X}_{1:T}) = \mathbf{h}^{s+1}_T, \tag{3}$$

where $\Gamma^{(s)}(\cdot)$ denotes the entire operation of C-TrellisNet with s layers.

3.2.2 Long-Term Temporal Correlation: P-TrellisNet. The C-TrellisNet introduced above takes only the outflows within the past few *close* time intervals (i.e., several hours) as input. However, *long-term temporal correlation* also should be considered in our MSP flow prediction problem.

Apparently, the values of the MSP flows among day-periodic time intervals (i.e., the same time intervals in the past days) are very close to each other. Such phenomenon indicates that MSP flows have an approximate long-term periodicity. To capture this long-term periodicity in MSP flows, we define the *day-periodicity data* as the MSP outflows in the same time intervals with the predicting target interval $t+1$ in the past D days, given as

$$\mathbf{P}_{1:D} = \left(\mathbf{x}_{t+1-Dm}, \mathbf{x}_{t+1-(D-1)m}, \cdots, \mathbf{x}_{t+1-m}\right),$$

where m denotes the number of time intervals in a day.

Furthermore, we observe that MSP flows between two adjacent days are discontinuous in time. In light of this, to better capture the long- and short-term correlation simultaneously, we further use another Periodicity TrellisNet (P-TrellisNet) after the C-TrellisNet and concatenate the day-periodicity data with the output of C-TrellisNet as P-TrellisNet's input. That is,

$$\widehat{\mathbf{X}}_{t+1} = \Gamma^{(p)}\left(\mathbf{P}_{1:D} \,||\, \Gamma^{(s)}(\mathbf{X}_{1:T})\right), \tag{4}$$

where s and p are the number of layers of the C-TrellisNet and P-TrellisNet respectively, and $\widehat{\mathbf{X}}_{t+1}$ is the eventual output of the CP-TrellisNets which is an n-dimensional vector consisting of each station i's $\widehat{X}_{i,t+1}$.

3.3 Dynamic Spatial Correlation: GC-TrellisNet

Clearly, the CP-TrellisNets defined above well captures MSP outflows' temporal correlation. However, considering temporal correlation alone is typically not enough to predict MSP outflows accurately. For example, when a special activity is going to be held somewhere in a city, the passenger outflows of nearby stations may increase dramatically in the future, which is hard to be learned from temporal features only.

To address this issue, we design a novel structure called GC-TrellisNet which captures not only the *spatial correlation* among metro stations, but also the *dynamics* of such correlation. Specifically, we place multiple DGCNs along the time steps and connect the output of each DGCN to a Dynamic TrellisNet (D-TrellisNet) in time-series order, as shown in Figure 3(b).

3.3.1 Diffusion Convolution on Metro Directed Graph. Clearly, the outflow of a station is influenced by the inflows of both the nearby and remote stations in the past few time intervals, and such influence is positively correlated with the transfer flow between two stations. Therefore, we propose a novel *transfer flow-based metric* to represent the spatial correlation among metro stations given in Definition 3.

Definition 3. *(Transfer Flow-Based Metric) Given the time interval t and the origin-destination station pair (i, j), we denote $\mathcal{A}_{ij}^t$ as the set of passengers who enter station i at some time instances, and exit station j in time interval t. The time window during which the passengers in $\mathcal{A}_{ij}^t$ enter station i is denoted as $[e_t, l_t]$. Based on the above notations, we define the transfer flow-based metric ξ_{ij}^t as*

$$\xi_{ij}^t = \frac{A_{ij}^t}{O_i^t}, \tag{5}$$

where A_{ij}^t denotes the cardinality of $\mathcal{A}_{ij}^t$ and O_i^t denotes the number of passengers entering station i in the time window $[e_t, l_t]$.

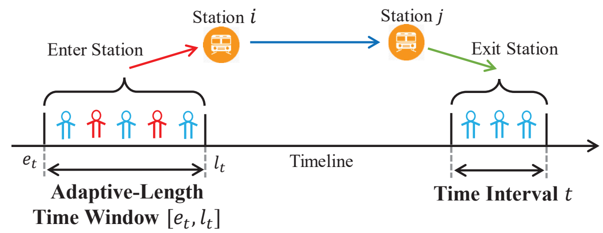


Figure 5: Illustration example of transfer flow-based metric, where a blue person represents a passenger exiting station j during time interval t who takes the metro from station i, and a red person represents a passenger who enters station i during time window $[e_t, l_t]$, but does not exit at station j.

Figure 5 illustrates a toy example to facilitate the understanding of the proposed metric. In this example, $A_{ij}^t = 3$, $O_i^t = 5$, and thus $\xi_{ij}^t = 3/5$. Note that the length of the time interval $[e_t, l_t]$ in Definition 3 is adaptive w.r.t. the time interval t. With such adaptive-length time window, the metric can represent the spatial correlation between any pair of origin-destination stations (i, j), regardless of their distance. Then, with the metric defined in Definition 3, we model the whole metro system as a *metro directed graph* given in Definition 4.

Definition 4. *(Metro Directed Graph) At each time interval t, we model the whole metro system as a weighted directed graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathbf{M}_t)$, where $\mathcal{V} = \{1, 2, \cdots, n\}$ denotes the vertex set consisting of all metro stations, $\mathcal{E}_t$ denotes the edge set, and $\mathbf{M}_t = \{a_{ij}^t\} \in [0,1]^{n\times n}$ denotes the adjacency matrix with ξ_{ij}^t's serving as the edge weights, i.e.,*

$$a_{ij}^t = \begin{cases} \xi_{ij}^t, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}. \tag{6}$$

By Definition 4, the metro directed graph models the metro system into a graph structure with adaptive edge weights, which we use as input to the *graph convolution network (GCN)* to capture the spatial correlation among MSP flows. According to [36], GCNs could be either spectral-based or spatial-based. The spectral-based models only apply to undirected graph due to the symmetry requirement of Laplacian matrix factorization, and are thus not suitable for our metro directed graph. Inspired by [12, 16], we integrate the diffusion process into spatial-based graph convolution and employ the *diffusion graph convolution network (DGCN)* on the metro directed graph to capture the spatial correlation of MSP flows.

More specifically, to predict the MSP outflow at time interval $t+1$, each DGCN in GC-TrellisNet takes an n-dimensional vector $\mathbf{Y}_{1:n,t}$ consisting of the n stations' MSP inflows at time interval t

as the input node features, given as

$$\mathbf{Y}_{1:n,t} = \left(f^{\text{in}}_{1,t}, f^{\text{in}}_{2,t}, \cdots, f^{\text{in}}_{n,t}\right)^{\text{tr}},$$

where $(\cdot)^{\text{tr}}$ represents the transpose operation. Then, we define the diffusion graph convolution in DGCN as

$$\Psi(\mathbf{M}_{t+1}, \mathbf{Y}_{1:n,t}) = \mathbf{M}_{t+1} \cdot \mathbf{Y}_{1:n,t} \circ \theta_w, \tag{7}$$

where $\Psi(\cdot)$ denotes the defined diffusion graph convolution, $\circ$ denotes the element-wise product, and θ_w represents the model parameters of DGCN. In fact, DGCN models the information propagation on the metro directed graph by a diffusion process that transfers the input node feature to one of its neighboring nodes with a certain transition probability characterized by the transfer flow-based metric.

3.3.2 Temporal Convolution on DGCNs. Note that the calculation defined by Equation (7) is carried out at time interval t, but requires the adjacency matrix $\mathbf{M}_{t+1}$ which is not available at time interval t. To address this issue, GC-TrellisNet uses a D-TrellisNet on top of multiple DGCNs placed in time-series order, which captures the dynamics of the adjacency matrix $\mathbf{M}_t$.

Specifically, GC-TrellisNet uses $s + 1$ DGCNs in parallel, and to predict the MSP outflow in time interval $t + 1$, we feed into each DGCN respectively the adjacency matrices $\mathbf{M}_{t-s}, \mathbf{M}_{t-s+1}, \cdots, \mathbf{M}_t$ but the same node feature sequence $\mathbf{Y}_{1:n,t}$. Then, the D-TrellisNet takes the output of each DGCN along the time steps as its input sequence, and thus our proposed GC-TrellisNet in fact carries out the operation $\Gamma^{(r)}(\cdot)$, given as

$$\widehat{\mathbf{Y}}_{t+1} = \Gamma^{(r)}\left(\Psi(\mathbf{M}_{t-s}, \mathbf{Y}_{1:n,t}), \cdots, \Psi(\mathbf{M}_t, \mathbf{Y}_{1:n,t})\right), \tag{8}$$

where $\widehat{\mathbf{Y}}_{t+1}$ denotes the output of the D-TrellisNet with r layers.

3.4 Bridge and Prediction

As aforementioned, STP-TrellisNets integrates two different parts, i.e., the temporal module CP-TrellisNets and the spatial module GC-TrellisNet. Specifically, as shown in Figure 3(c), it merges the output of the CP-TrellisNets with that of GC-TrellisNet by a fusion (referred to as Bridge), and uses the merged value as the predicted MSP outflow vector $\widehat{\mathbf{Z}}_{t+1}$. That is, the final output of the STP-TrellisNets satisfies that

$$\widehat{\mathbf{Z}}_{t+1} = \tanh\left(\mathbf{W}_x \circ \widehat{\mathbf{X}}_{t+1} + \mathbf{W}_y \circ \widehat{\mathbf{Y}}_{t+1}\right),$$

where $\mathbf{W}_x$ and $\mathbf{W}_y$ are the learnable parameters that measure how the results of temporal correlation learning and dynamic spatial correlation learning affect MSP flow prediction, and $\circ$ denotes again the element-wise multiplication.

In the training process, we aim to minimize the mean square error (MSE) represented by the loss function

$$L(\mathbf{\Theta}) = \frac{1}{bn}\sum_{a=1}^{b}\sum_{i=1}^{n}\left(Z^a_{i,t+1} - \widehat{Z}^a_{i,t+1}\right)^2,$$

where b denotes the number of samples in a training batch and $\mathbf{\Theta}$ denotes all learnable parameters of the STP-TrellisNets.

Table 1: Detailed information of the evaluated datasets.

Properties	Datasets	
	Shenzhen	Hangzhou
# of Stations	165	81
# of Records	1.5 billion	70 million
Time span	6/1/2017 - 6/30/2017	1/1/2019 - 1/25/2019
Daily range	6:30-23:00	6:00-23:30

4 EXPERIMENTS

In this section, we introduce the datasets, data preprocessing, the baseline methods with which we compare STP-TrellisNets, the setup of our experiments and the corresponding experimental results.

4.1 Datasets

In this paper, we evaluate the performance of STP-TrellisNets on two large-scale real-world *automated fare collection (AFC)* datasets, namely Shenzhen and Hangzhou. The Shenzhen dataset is collected from the metro AFC system in the city of Shenzhen, China, consisting of about 1.5 billion AFC records from June 1st to June 30th, 2017. The Hangzhou dataset, made available for the Global Urban Computing AI Challenge 2019@TIANCHI[2], contains about 70 million AFC records in the city of Hangzhou, China from January 1st to January 25th, 2019. Detailed information of the two datasets are given in Table 1.

4.2 Data Preprocessing

4.2.1 ***MSP Inflow and Outflow.*** Each piece of the AFC record in our datasets contains features including *user card ID*, *metro station name*, *inbound or outbound label*, and *recording timestamp*. The following Table 2 shows an example of one piece of record in the datasets.

Table 2: Example of one piece of AFC record in the datasets.

Field	Value
User Card ID	685374531
Timestamp (ts)	20170604201430
In or Out (io)	In
Line Number	11
Station Name (sn)	Nanshan

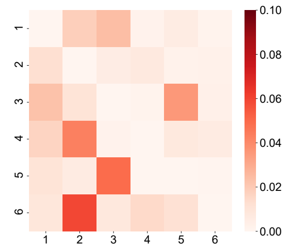
We use $\mathcal{D}$ to denote the set of AFC records in one dataset (i.e., Shenzhen or Hangzhou). Then, we count the number of inbounds and outbounds of each metro station i in each time interval t to obtain the MSP inflow $f^{\text{in}}_{i,t}$, such that

$$f^{\text{in}}_{i,t} = \sum_{r\in\mathcal{D}} \mathbb{1}\{r.sn = i \;\&\; r.io = \text{In} \;\&\; r.ts \in \text{interval } t\},$$

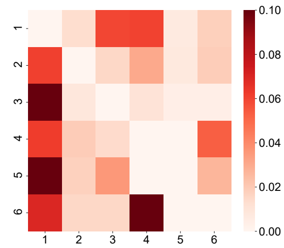
and the MSP outflow $f^{\text{out}}_{i,t}$, such that

$$f^{\text{out}}_{i,t} = \sum_{r\in\mathcal{D}} \mathbb{1}\{r.sn = i \;\&\; r.io = \text{Out} \;\&\; r.ts \in \text{interval } t\}.$$

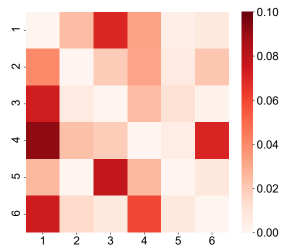
[2]https://tianchi.aliyun.com/competition/entrance/231708/information



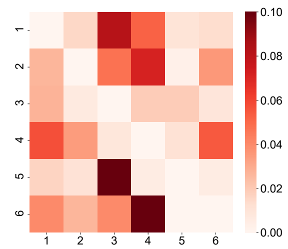
(a) 8:00-8:30.



(b) 12:00-12:30.



(c) 16:00-16:30.



(d) 22:00-22:30.

Figure 6: Adjacency matrices of 6 stations in 4 different time intervals within a day.

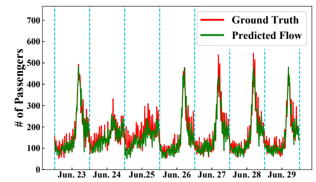
4.2.2 ***Metro Directed Graph.*** According to Definition 4, we obtain ξ_{ij}^{t} from our AFC datasets to construct the adjacency matrix $\mathbf{M}_t$ that corresponds to each time interval t. Note that the starting and ending time instances of the adaptive-length time window $[e_t, l_t]$ is obtained by tracking the user card ID in the records. Furthermore, we use the inflow of each metro station obtained above to construct the node feature vector $\mathbf{Y}_{1:n,t}$. Figure 6 shows the adjacency matrices in 4 different time intervals within one day with 6 stations indexed as Station 1, 2, $\cdots$, 6 which correspond respectively to Station Shenzhenbei, Shangmeilin, Wuhe, Longhua, Tanglang, and Shangtang in the city of Shenzhen, China. From this figure, we can observe that such adjacency matrices change over time, which shows the dynamic nature of the spatial correlation among stations.

4.3 Baseline Methods

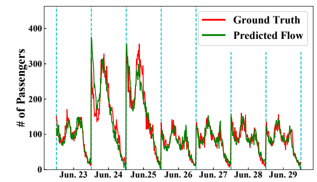
In this paper, we compare our STP-TrellisNets with the following baseline methods. For all the baseline methods, we use the outflows in the last few time intervals (i.e., $t - T, \cdots, t$) as the historical data to predict the outflow in the next time interval (i.e., $t + 1$).

(1) **Last-Value:** Last-Value returns the MSP outflow value in the last time interval of the historical data as the predicted outflow value of the future. (2) **HA:** Historical Average. (3) **VAR [37]:** A multivariate generalization of autoregression, which captures the correlation among multiple time series. (4) **MLP:** MLP consists of multiple layers of neurons in a feed-forward way. We build an MLP model with a single hidden layer of 256 units. (5) **LSTM [38]:** A class of RNN that is capable of capturing temporal correlation of time-series data. (6) **ConvLSTM [39]:** a variant of LSTM, in which convolutional structure[3] is used to further capture spatial correlation of data. (7) **DCRNN [12]:** A spatial-temporal deep learning model, which combines diffusion convolutional network and GRU-based RNN to capture the spatial correlation and temporal correlation of traffic data. (8) **STGCN [7]:** A spatial-temporal deep learning framework, which formulates the problem on graphs and builds the model with complete convolutional structures. (9) **Graph Wavenet [14]:** A graph neural network model, which captures the spatial correlation with a diffusion convolution layer and learns the temporal correlation using generic TCN.

[3]Because the metro system cannot be treated as an image-like structure, we treat the metro line as a 1D grid and use 1D CNN to capture the spatial correlation in our experiment.

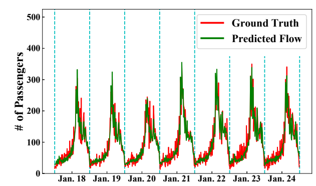


(a) **Station Nanshan.**

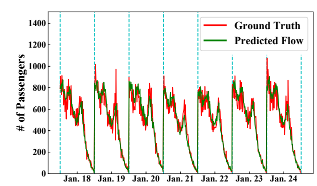


(b) **Station Shaoniangong.**

Figure 7: Comparison of the MSP outflows predicted by STP-TrellisNets and the ground truth values of the two selected stations in the Shenzhen dataset.



(a) **Station Xiasha Jiangbin.**



(b) **Station Longxiangqiao.**

Figure 8: Comparison of the MSP outflows predicted by STP-TrellisNets and the ground truth values of the two selected stations in the Hangzhou dataset.

4.4 Experimental Setup

In our experiments, we employ three widely-adopted metrics, namely Mean Absolute Error(MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to evaluate the performances of STP-TrellisNets and the baseline methods.

The evaluation datasets are split into training, validation, and testing datasets with the ratio of $7 : 1 : 2$. For CP-TrellisNets, the input sequence lengths of C-TrellisNet and P-TrellisNet are adjusted to cover two fixed historical time spans, 6 hours and 3 days, which correspond to the short- and long-term historical data respectively. For GC-TrellisNet, we implement 3 DGCNs in parallel and use the 3 adjacency matrices of the past 3 time intervals as inputs.

4.5 Performance Evaluation

4.5.1 ***Comparison with Ground Truths.*** In Figure 7 and Figure 8, we visualize the comparison between the MSP outflows in 10 minutes predicted by STP-TrellisNets and the ground truth values of four representative metro stations. From the figures, we could observe that the MSP outflows predicted by STP-TrellisNets are fairly close to the ground truth values.

4.5.2 ***Comparison with Baseline Methods.*** We compare our STP-TrellisNets with all baseline methods with 3 different lengths of time intervals, namely 10, 15, and 30 minutes, respectively. Table 3 summarizes the performances of STP-TrellisNets and all baseline methods on the Shenzhen and Hangzhou datasets. Next, we will

Table 3: Performance comparison with baseline methods on both datasets of 3 different time intervals length.

Datasets	Models	10 minutes			15 minutes			30 minutes		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
Shenzhen	Last-Value	22.92	44.33	33.83	30.25	67.48	28.15	62.86	157.68	31.05
	HA	39.30	84.23	58.30	49.24	121.29	34.26	105.91	225.75	47.99
	VAR	20.46	42.05	29.92	29.09	68.20	20.28	58.63	147.89	24.28
	MLP	18.02	30.81	27.61	26.35	49.10	16.40	45.57	78.33	19.81
	LSTM	15.99	26.01	22.17	28.35	49.79	16.64	53.15	95.77	19.34
	ConvLSTM	16.99	28.19	23.64	21.68	35.54	13.87	32.35	52.81	12.88
	DCRNN	15.66	27.40	21.86	22.00	40.66	13.92	32.90	56.76	12.84
	STGCN	15.61	26.87	20.84	17.73	28.65	17.75	30.30	50.72	16.28
	Graph WaveNet	15.05	25.94	20.21	18.19	31.38	16.92	29.76	51.09	16.52
	STP-TrellisNets	**13.46**	**22.03**	**14.78**	**16.40**	**26.65**	**13.10**	**25.11**	**41.72**	**12.74**
Hangzhou	Last-Value	31.02	54.96	26.28	41.87	79.51	23.20	87.82	179.81	23.36
	HA	34.52	76.47	28.81	48.39	110.22	26.52	90.58	212.21	24.03
	VAR	32.20	58.68	24.48	34.97	80.69	22.42	81.49	207.74	25.14
	MLP	31.50	46.79	25.26	31.14	60.26	27.92	66.17	117.13	24.21
	LSTM	28.02	37.25	22.63	28.22	42.63	21.20	70.76	118.50	17.49
	ConvLSTM	26.85	34.85	21.55	27.72	38.18	19.56	51.33	74.65	17.69
	DCRNN	26.20	35.78	20.65	26.10	35.32	16.06	55.39	63.14	14.04
	STGCN	21.86	35.51	18.29	24.78	40.06	14.29	38.74	59.15	12.05
	Graph WaveNet	19.53	32.38	17.22	23.59	39.12	14.00	36.82	61.63	11.55
	STP-TrellisNets	**18.37**	**28.86**	**16.61**	**21.94**	**33.99**	**13.00**	**34.37**	**55.56**	**9.56**

discuss some observations of the results and the rationale behind them.

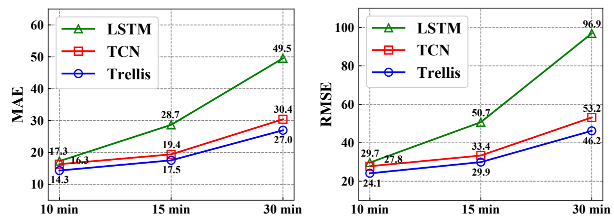
Firstly, it is easily observable that STP-TrellisNets achieves the best performance in terms of all performance metrics. Secondly, we can observe that performances of all methods in terms of MAE and RMSE degrade as the length of time interval increases. This is because a longer time interval leads to larger peak value and higher fluctuation of MSP flow within one time interval. Moreover, the prediction performance by Last-Value is undesirable, which reflects the massive dynamic changes of MSP flow values and proves the necessity of proposing effective MSP flow prediction method.

As HA and VAR assume the linear temporal correlation among time intervals, they fail to capture the nonlinear correlation, and thus have worse performances on the prediction of MSP flows. Although nonlinear activation functions empower MLP and LSTM the ability to learn the nonlinear temporal correlation, MLP and LSTM do not capture spatial-temporal correlation in an explicit manner. ConvLSTM employs conventional convolutional structure to model the spatial correlation, which is not completely applicable to non-Euclidean graph-structured data in our problem. STGCN employs graph convolution network to capture the spatial correlation of MSP outflows but fail to consider the dynamics of such correlation.

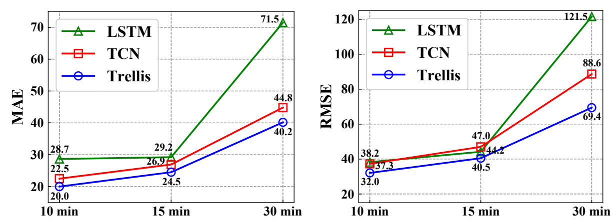
Among DCRNN, Graph Wavenet and our STP-TrellisNets, all of which exploit the diffusion convolution to capture the spatial correlation while exploring different methods to extract the temporal correlation, our proposed STP-TrellisNets achieves the best performance. The reasons behind are two-fold. On one hand, compared to DCRNN and Graph Wavenet, our STP-TrellisNets are more effective in capturing the temporal correlation because of its TrellisNet structure and a proper consideration of the long-term periodicity of MSP outflows. On the other hand, our STP-TrellisNets are better than the other two models on capturing the dynamic spatial correlation of MSP outflows, because it uses a novel transfer flow-based metric to represent the spatial correlation and further integrates the diffusion convolution with TrellisNet to capture the dynamics of such correlation.

4.5.3 ***Comparison of Models for Capturing Temporal Correlation.*** In this part, we evaluate the performance of TrellisNet for capturing the temporal correlation of MSP outflows. We compare the prediction errors of the MSP outflow in the next time interval using the historical data in the past 24 hours among three different models, namely 1) LSTM, 2) generic TCN with 4 convolution layers and dilation factors $(1, 1, 2, 2)$ in each layer, and 3) the C-TrellisNet introduced in Section 3.2.1 with 4 layers. Figure 9 summarizes the comparison results of the three models on the Shenzhen and Hangzhou datasets. From this figure, we can see that C-TrellisNet, which inherits the advantages of TCN and RNN on capturing the temporal correlation, indeed outperforms the other two models.

4.5.4 ***Necessity of Handling the Discontinuity of MSP Flows.*** As we have introduced before, we propose the CP-TrellisNets architecture to handle the discontinuity between long-term and short-term MSP flows, because the MSP flows between two adjacent days are discontinuous in time. To justify the necessity of such design choice, we compare the performance between STP-TrellisNets and a variant of STP-TrellisNets, namely *cSTP-TrellisNets*, which omits the discontinuity of the MSP flows between two adjacent days and employs only one TrellisNet to learn both the short- and long-term temporal correlation of MSP flows. Figure 10 shows that the STP-TrellisNets yields significantly better performance of MSP

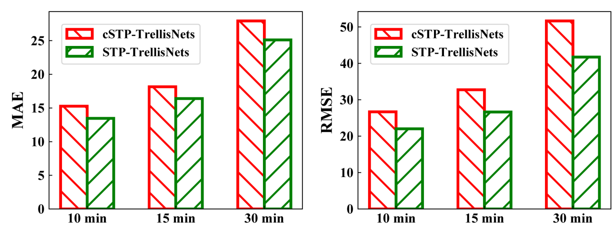


(a) **LSTM vs. generic TCN vs. TrellisNet on the Shenzhen dataset**

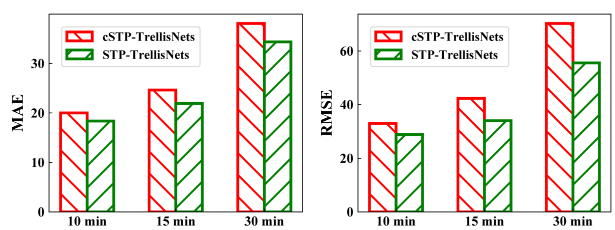


(b) **LSTM vs. generic TCN vs. TrellisNet on the Hangzhou dataset**

Figure 9: The prediction performance comparison among LSTM, generic TCN, and C-TrellisNet.



(a) **STP-TrellisNets vs. cSTP-TrellisNets on the Shenzhen dataset**



(b) **STP-TrellisNets vs. cSTP-TrellisNets on the Hangzhou dataset**

Figure 10: The prediction performance comparison between the STP-TrellisNets and cSTP-TrellisNets.

flow prediction on both the Shenzhen and Hangzhou datasets than cSTP-TrellisNets. Such results indicate the necessity of handling the discontinuity of MSP flows, and verify the effectiveness of our CP-TrellisNets on this task.

4.5.5 *Comparison with Architectural Variants.*

In our experiments, to further verify the effectiveness of the proposed architecture of STP-TrellisNets, we also vary its structure, and evaluate the performances of the following architectural variants.

- **C-TrellisNet:** In this variant, we employ only the C-TrellisNet structure which merely considers the short-term temporal correlation of MSP outflows.
- **CP-TrellisNets:** In this variant, we retain the CP-TrellisNets structure in STP-TrellisNets. Clearly, the long- and short-term temporal correlation is captured by this variant without considering the spatial correlation.
- **CP-TrellisNets+Single DGCN:** In this variant, apart from CP-TrellisNets, we use a single DGCN without any TrellisNets connected to its output, which apparently cannot capture the dynamics of spatial correlation. Furthermore, the input adjacency matrix of the DGCN at time interval t is set to be $\mathbf{M}_t$.

Table 4: Comparison with architectural variants.

Models	MAE	RMSE	MAPE (%)
C-TrellisNet	37.43	63.56	10.10
CP-TrellisNets	36.65	62.65	9.96
CP-TrelliNets+Single DGCN	36.19	61.25	9.77
STP-TrellisNets	**34.37**	**55.56**	**9.56**

Table 4 shows their performances of MSP flow prediction in 30 minutes on the Hangzhou dataset. We can easily observe that STP-TrellisNets achieves a much better performance than C-TrellisNet and CP-TrellisNets, because they cannot capture spatial correlation. Besides, CP-TrellisNets+Single DGCN's mediocre performance demonstrates that the effectiveness and necessity of capturing the dynamics of spatial correlation in the spatial module of STP-TrellisNets.

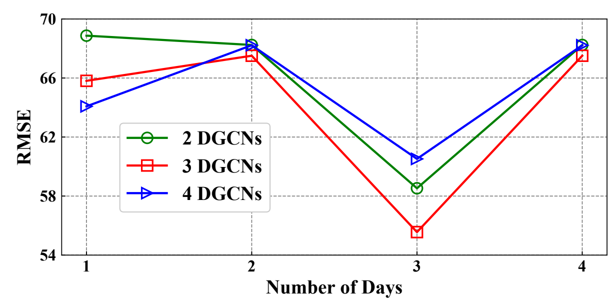


Figure 11: Joint influence of CP-TrellisNets' input length and the number of DGCNs.

4.5.6 *Joint Influence of CP-TrellisNets' Input Length and the Number of DGCNs in GC-TrellisNet.*

In this set of experiments, we evaluate the MSP flow prediction performance in 30 minutes on the Hangzhou dataset, with various number of days of long-term periodic data and number of DGCNs in GC-TrellisNet in order to explore the best parameter setting of STP-TrellisNets. As shown in Figure 11, our choice of using 3 DGCNs outperforms the structures with 2 and 4 DGCNs in terms of RMSE. Furthermore, our STP-TrellisNets structure with 3 DGCNs achieves the best performance by using the past 3 days' periodic MSP flows as inputs to the P-TrellisNets.

5 CONCLUSION AND FUTURE WORK

In this paper, we address the MSP flow prediction problem from both the spatial and temporal perspectives by proposing a novel deep learning framework STP-TrellisNets. Specifically, from the **temporal** perspective, we adopt TrellisNet to handle the long-range input historical data of MSP flows, and propose CP-TrellisNets consisting of two TrellisNets in serial for capturing the long- and short-term temporal correlation of MSP flows. The necessity and effectiveness of such design choices are validated by our various experimental results. Especially, our comparison, given in Section 4.5.4, between the STP-TrellisNets with cSTP-TrellisNets which has only one TrellisNet as its temporal module indicates that our CP-TrellisNets structure is effective in jointly capturing the long-term and short-term temporal correlation. From the **spatial** perspective, we propose the GC-TrellisNet framework, which uses a transfer flow-based metric to characterize the spatial correlation among MSP flows, and employs multiple DGCNs along the time steps with their outputs fed to a TrellisNet to capture the dynamics of such spatial correlation. Such design choices that essentially integrate TrellisNet with graph convolution are also shown to be effective by our experimental results that compare STP-TrellisNets with various architectural variants in Section 4.5.5. Jointly from the **spatial-temporal** perspective, our STP-TrellisNets that fuses the outputs of the CP-TrellisNets and GC-TrellisNets as the final predicted MSP flows outperforms all the baseline approaches in our extensive experiments with two large-scale real-world AFC datasets.

Although the STP-TrellisNets is tailored in various aspects to deal with the special features of MSP flows, we envision that it is, or at least some of its components are, promising to be generalized to other prediction tasks with only minor adaptations. For future work, we would investigate the generalize ability of the STP-TrellisNets from the following two aspects: (1) exploiting the CP-TrellisNets for time-series prediction scenarios which also exhibit long- and short-term temporal correlation; (2) augmenting the feature engineering process and graph convolution operations of the GC-TrellisNet to capture the dynamic spatial correlation of other graph-structured data which also possess such dynamics.

ACKNOWLEDGMENTS

This work was supported in part by National Key R&D Program of China 2018AAA0101200, in part by NSF China under Grant (No. 61960206002, 61902244, 62041205, 61822206, 61532012, 61942204), in part by Major Scientific Research Project of Zhejiang Lab (No. 2019DB0ZX01), in part by Shanghai Municipal Science and Technology Commission Grant 19YF1424600, in part by the Science and Technology Innovation Program of Shanghai (Grant 18XD1401800), in part by Shanghai Key Laboratory of Scalable Computing and Systems.

REFERENCES

[1] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transportation Research Record*, 2008.
[2] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *WSDM*, 2018.
[3] B. Shen, X. Liang, Y. Ouyang, M. Liu, W. Zheng, and K. M. Carley, "Stepdeep: A novel spatial-temporal mobility event prediction framework based on deep neural network," in *KDD*, 2018.
[4] T.-y. Fu and W.-C. Lee, "Deepist: Deep image-based spatio-temporal network for travel time estimation," in *CIKM*, 2019.
[5] L. Bai, L. Yao, S. S. Kanhere, X. Wang, W. Liu, and Z. Yang, "Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction," in *CIKM*, 2019.
[6] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng, "City-wide traffic volume inference with loop detector data and taxi trajectories," in *SIGSPATIAL*, 2017.
[7] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018.
[8] A. Zonoozi, J.-j. Kim, X.-L. Li, and G. Cong, "Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns." in *IJCAI*, 2018.
[9] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *AAAI*, 2019.
[10] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling," in *KDD*, 2019.
[11] C. Chen, K. Li, S. G. Teo, X. Zou, K. Wang, J. Wang, and Z. Zeng, "Gated residual recurrent graph neural networks for traffic prediction," in *AAAI*, 2019.
[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.
[13] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "Gstnet: Global spatial-temporal network for traffic flow prediction," in *IJCAI*, 2019.
[14] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*, 2019.
[15] S. Bai, J. Z. Kolter, and V. Koltun, "Trellis networks for sequence modeling," in *ICLR*, 2019.
[16] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *NIPS*, 2016.
[17] 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 *AAAI*, 2018.
[18] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *AAAI*, 2019.
[19] B. Du, H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li, "Deep irregular convolutional residual lstm for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
[20] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *AAAI*, 2017.
[21] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[22] L. Bai, L. Yao, S. Kanhere, X. Wang, Q. Sheng *et al.*, "Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *IJCAI*, 2019.
[23] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI*, 2019.
[24] C. Park, C. Lee, H. Bahng, K. Kim, S. Jin, S. Ko, J. Choo *et al.*, "Stgrat: A spatio-temporal graph attention network for traffic forecasting," in *AAAI*, 2019.
[25] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *KDD*, 2019.
[26] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *AAAI*, 2019.
[27] M. Wang, B. Lai, Z. Jin, Y. Lin, X. Gong, J. Huang, and X. Hua, "Dynamic spatio-temporal graph-based cnns for traffic prediction," *arXiv preprint arXiv:1812.02019*, 2018.
[28] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Zheng, and C. Kirsch, "Network-wide crowd flow prediction of sydney trains via customized online non-negative matrix factorization," in *CIKM*, 2018.
[29] J. Zhao, Q. Qu, F. Zhang, C. Xu, and S. Liu, "Spatio-temporal analysis of passenger travel patterns in massive smart card data," *IEEE Transactions on Intelligent Transportation Systems*, 2017.
[30] G. Sun, Y. Xiong, and Y. Zhu, "How the passengers flow in complex metro networks?" in *SSDBM*, 2017.
[31] Y. Liu, Z. Liu, and R. Jia, "Deeppf: A deep learning based architecture for metro passenger flow prediction," *Transportation Research Part C: Emerging Technologies*, 2019.
[32] Y. Ning, Y. Huang, J. Li, Q. Liu, D. Yang, W. Zheng, and H. Liu, "St-drn: Deep residual networks for spatio-temporal metro stations crowd flows forecast," in *IJCNN*, 2018.
[33] L. Tang, Y. Zhao, J. Cabrera, J. Ma, and K. L. Tsui, "Forecasting short-term passenger flow: An empirical study on shenzhen metro," *IEEE Transactions on Intelligent Transportation Systems*, 2018.
[34] S. Hao, D.-H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," *Transportation Research Part C: Emerging Technologies*, 2019.
[35] Z. Li, N. D. Sergin, H. Yan, C. Zhang, and F. Tsung, "Tensor completion for weakly-dependent data on graph for metro passenger flow prediction," in *AAAI*, 2020.
[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[37] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-Plus®*, 2006.
[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.
[39] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015.