# MSTNN: A Graph Learning Based Method for the Origin-Destination Traffic Prediction

Chenming Yang[†], Zhiheng Zhou[†‡], Hui Wen[*], and Liang Zhou[†‡],
[†]National Key Laboratory of Science and Technology on Communications,
[*]Department of Computer Science and Engineering,
[‡]Center for Intelligent Networking and Communications,
University of Electronic Science and Technology of China, Chengdu, P. R. China
Email: chenmingyang@std.uestc.edu.cn, zhzhou@uestc.edu.cn, huiwen@std.uestc.edu.cn, lzhou@uestc.edu.cn

***Abstract*—Accurate origin-destination traffic prediction (ODTP) is a persistent problem in network management. Due to the structural nature of networks, the spatial correlations are critical for effective perdition. In this paper, the ODTP problem is built onto the graph domain by mapping OD traffic into graph-structured data involving the topology information. Moreover, due to the flow characteristics of OD traffic, the complex spatial-temporal (ST) correlations should not be limited at one-hop neighbors or consecutive time steps. To benefit from this observation, we propose a novel graph learning based method, called multi-scale spatial-temporal graph neural network (MSTNN). In MSTNN, spatial and temporal extractors are designed to capture multi-scale spatial and temporal correlations. Specifically, the spatial extractor employs the graph attention mechanism to capture the time-varying spatial correlations of nodes and their multi-hop neighbors. In the temporal extractor, we design gated dilated convolution layers with different dilation factors, each of which represents a different granularity of time, to exploit the multi-scale temporal correlations in both adjacent and non-adjacent time steps. After cascading the spatial and temporal extractors, the multi-scale ST correlations are weighted fused. Simulations on two real-world datasets show that MSTNN outperforms existing approaches that work well in various prediction tasks.**

***Index Terms*—network management, OD traffic prediction, graph learning, spatial-temporal correlation**

## I. Introduction

Origin-destination (OD) traffic is vital in network QoS management [1], such as traffic engineering [2], routing planning [3], and anomaly detection [4]. In these applications, the decisions can be made more efficient and effective by using estimated future OD traffic. However, making timely accurate predictions of OD traffic is a difficult problem, due to the high nonlinearity and intricate patterns of OD flows.

In general, the OD traffic prediction (ODTP) problem is converted to the traffic matrix (TM) prediction problem [2]. The existing TM predictors are divided into two categories: single flow predictors (SFPs) and multi flows predictors (MFPs). SFPs such as ARIMA [5] predict each single OD flow based on its historical data. In such a way, traffic of different OD flows is handled independently, resulting in loss of spatial correlations of different flows. MFPs, such as Kalman filters [2] and CNN [6], are proposed to take multiple flows into account. However, these methods are unaware of the ST correlations related to the network topology and thus cannot perform well consistently.

In order to leverage the topology-based ST correlations, we model the ODTP problem to a node attribute prediction problem on the graph. In this paper, the attribute of a node is the traffic volumes of the OD flows originated from the node. Therefore, the OD traffic data can be mapped onto the graph domain because of the structural nature of networks. The critical problem is then how to extract ST correlations from such graph-structured OD traffic data.

Recently, various spatial-temporal neural networks (STNNs) are proposed to analyze graph-structured data efficiently and applied to various prediction tasks [7]. For instance, [8] introduces the diffusion convolution operator to capture spatial correlations and uses gated recurrent units to capture temporal dependencies; STGCN [9] interleaves 1-dimension CNN with Graph Convolution Network to learn from graph data series.

However, most of STNNs cannot recognize time-varying spatial correlations due to the use of the graph convolution [10]. Moreover, the temporal correlation extractors using Recurrent Neural Networks (RNNs) [11] or 1-dimension CNN overlook the temporal patterns in non-adjacent time steps, which can reveal the long-term trend of time series.

In this paper, we propose a novel graph learning based method, namely multi-scale spatial-temporal graph neural networks (MSTNN), to overcome the above limitations. To do so, we design multi-scale temporal (MST) sub-blocks and spatial (MSS) sub-blocks. In a MST sub-block, the gated dilated convolution (Gated DC) layers are developed to recognize the multi-scale temporal correlations in both adjacent and non-adjacent time steps. In a MSS sub-block, we employ the graph attention mechanism [12] on multi-hop neighbors of some node to learn the time-varying spatial correlations of OD flows. We then design a multi-scale ST (MSST) block, including cascaded MST sub-blocks and MSS sub-blocks, to fuse the corresponding multi-scale ST correlations with the coefficients learned during training. To the best of our knowledge, it is the first time to introduce graph learning based method to the ODTP problem. Evaluation results on datasets Abilene [13] and GÉANT [14] indicate that our approach is more effective than existing methods in prediction

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

tasks regarding prediction length and network scale. The main contributions of this paper are summarized as follows:

- We treat OD traffic as graph-structured data and model the ODTP problem on graphs to benefit from topology-based ST correlations of OD flows.
- We explore graph learning techniques to handle graph-structured OD traffic data and then design a novel graph learning based predictor MSTNN, which can extract multi-scale ST correlations of the OD traffic.
- Experiments are carried out on two real-world datasets to show the better performance of the proposed predictor.

This paper is organized as follows. Section II introduces the ODTP problem on graphs and briefly describe existing extractors of spatial and temporal correlations. Section III presents the details of the proposed MSTNN. Section IV reports the experiment results, along with the discussions. Finally, the conclusions are made in Section V.

## II. The ODTP Problem on Graphs

### A. Problem Statement

A network can be represented as a graph $\mathcal{G}(\mathcal{N}, \mathbf{W})$, where $\mathcal{N}$ is the set of $N$ nodes; $\mathbf{W} := [W_{ij}] \in \{0,1\}^{N\times N}$ is the adjacency matrix. Network OD traffic at time $t$ can be represented by a traffic matrix $\mathbf{X}(t) := [x_{ij}(t)] \in \mathbb{R}^{N\times N}$, where $x_{ij}(t)$ is the traffic volume of the OD flow originated from node $i$ and destined for node $j$ at time $t$. Let $\mathbf{X}^-(t;T) = (\mathbf{X}(t-T+1), \ldots, \mathbf{X}(t))$ be the OD matrices in previous $T$ time steps and $\mathbf{X}^+(t;M) = (\mathbf{X}(t+1), \ldots, \mathbf{X}(t+M))$ be the OD matrices in future $M$ time steps. Traditional MFPs to solve the ODTP problem is to build a predictor with input $\mathbf{X}^-(t;T)$ to predict $\mathbf{X}^+(t;M)$.

However, it can be easily seen from the definition of $\mathbf{X}^-(t;T)$ that these matrices do not include the topology information of the network. This causes traditional MFPs cannot extract spatial correlations, which is proven to be crucial for the ODTP problem [9].

To leverage the graph nature of the network, we build the ODTP problem on the graph $\mathcal{G}$ as follows. Let $\mathbf{x}_i(t)$ be the $i$-th row of $\mathbf{X}(t)$. By labelling node $i$ with $\mathbf{x}_i(t)$, $i = 1, 2, \ldots, N$, OD traffic data are mapped onto graph $\mathcal{G}$, forming the graph-structured OD traffic data $\mathcal{G}(t) = \mathcal{G}(\{\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t)\}, \mathbf{W})$. Let $\mathcal{G}^-(t;T) = (\mathcal{G}(t-T+1), \mathcal{G}(t-T+2), \ldots, \mathcal{G}(t))$, as shown in Fig. 1. The ODTP problem on $\mathcal{G}$ is to build a predictor using historical graph data $\mathcal{G}^-(t;T)$ as inputs to predict $\mathcal{X}^+(t;M)$.

### B. Extractors of Spatial Correlations

Spatial [15] and spectral graph convolution [16] are two popular ways of analyzing graph-structured data.

Spatial graph convolution rearranges network nodes into certain grid forms and combines the features of both the nodes themselves and their neighbors. For instance, Equation 1 is a well-known method to handle graph-structured data [15],

$$\mathbf{r}_i^l = \sigma\left(\sum_{j\in\mathcal{N}_i} \mathbf{\Theta}^{l-1}\mathbf{r}_j^{l-1}\right), \tag{1}$$

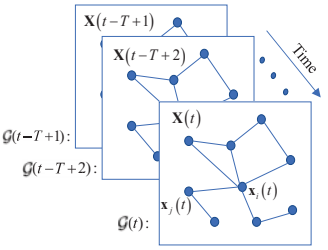



Fig. 1: Graph-structured OD traffic data.

where $\mathbf{r}_i^l$ is the hidden representation of node $i$ in layer $l$; $\mathcal{N}_i$ is the set of one-hop neighbors of node $i$; $\sigma(\cdot)$ is a non-linear function; $\mathbf{\Theta}^l$ is the kernel in layer $l$. However, Eq. 1 cannot handle time-varying spatial correlations, and thus may not perform well on temporal graphs.

On the other hand, spectral graph convolution implements the convolution in the spectral domain by graph Fourier transforms [16]. Because the convolution kernel has $N^2$ elements, the original spectral graph convolution suffers from high computational complexity. In addition, it is unable to distinguish the contributions of neighbors in different hops, when generating the hidden representation of nodes data. To tackle these problems, authors in [10] proposes a 1st-order approximation to the hidden representation of nodes data,

$$\mathbf{r}^l \approx \mathbf{\Theta}(\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}})\mathbf{r}^{l-1} \tag{2}$$

where $\mathbf{\Theta}$ is a convolution kernel; $\tilde{\mathbf{A}}$ is the presupposed weighted adjacency matrix of $\mathcal{G}$; $\mathbf{I}_n$ is an identity matrix, and $\mathbf{D} \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. Using this approximation, [9] shows the state-of-the-art performance in the traffic forecasting task. However, Equation 2 only considers the spatial correlations within 1st-hop neighbors, which limits its capability.

Besides, both Eq. 1 and 2 focus on recognizing static spatial patterns, while spatial correlations could be time-varying. Lossing dynamic of spatial correlations further degrades the capability of the predictor in realistic networks.

To overcome the above limitations, we design a multi-scale spatial correlations extractor that employs the masked graph attention mechanism [12]. A masked graph attention operator can acquire time-varying spatial correlations by computing self-attention coefficients among different nodes. Moreover, the operators can be stacked to extract multi-hop spatial correlations. The details are described in Section III-B.

### C. Extractors of Temporal Correlations

Temporal pattern recognition is another primary task of the ODTP problem. In the past decades, RNNs [11] are widely applied in time series analysis. However, RNNs suffer from time-consuming iterations, complex gate mechanisms, and are insensitive to dynamic changes.

To improve the time efficiency, [9] proposes gated sequential CNN (Gated CNN), which uses 1-dimension CNN to capture temporal patterns of traffic flows. As a result of using the standard convolution, Gated CNN can only extract

temporal correlations in adjacent time steps and misses the temporal patterns in non-adjacent time steps. To identify multi-scale temporal correlations, we propose the gated dilated convolution (Gated DC) layers that can capture temporal features in both adjacent and non-adjacent time steps. The details of Gated DC layers are illustrated in Section III-C.

## III. Multi-scale Spatial-temporal Graph Neural Network

In this section, a novel graph learning based method, MSTNN, is presented to solve the graph-based ODTP problem. MSTNN is a supervised learning model, in which training data, i.e., historical OD traffic, are sliced over a sliding window with length $T+M$. The first $T$ TMs $\mathbf{X}^{-}(t;T)$ and the adjacency matrix $\mathbf{W}$ are the inputs to generate the predictions $\tilde{\mathbf{X}}^{+}(t;M) = (\tilde{\mathbf{X}}(t+1), \ldots, \tilde{\mathbf{X}}(t+M))$, while the last $M$ TMs $\mathbf{X}^{+}(t;M)$ are treated as ground truths.

### A. Network Architecture

Figure 2 illustrates an example of the structure of MSTNN from a multi-level perspective. As shown in Fig. 2(a), MSTNN consists of two stacked MSST blocks. The residual connection is applied in each MSST block. The final fully-connected (FC) layer maps outputs of the last MSST block to predictions $\tilde{\mathbf{X}}(t;M) = (\tilde{\mathbf{X}}(t+1), \ldots, \tilde{\mathbf{X}}(t+M))$.

The MSST block is shown in Fig. 2(b) with input $\mathbf{R}^{-}_{in}(t;T^{in}_{R})$ and output $\mathbf{R}^{-}_{out}(t;T^{out}_{R})$, while the structure of MSS and MST sub-blocks are shown in Fig. 2(c) and (d) respectively. The MSST block is composed of $n$ MST sub-blocks implementing the gated dilated convolution. Each MST sub-block is followed by $m$ MSS sub-blocks employing the graph attention mechanism. It is to be noticed that the number of MSST blocks, MST, and MSS sub-blocks are hyper-parameters. Then the MSTNN can be denoted by MSTNN$(m,n)$, and we will illustrate its relationship with spatial and temporal scales in Section III-C(D)

The loss function of training MSTNN is,

$$Loss = \sum_{m=1}^{M} \left\| \tilde{\mathbf{X}}(t+m) - \mathbf{X}(t+m) \right\|_{F}, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm.

The detailed procedures of the spatial and temporal extractions are described shortly, followed by the design of fusing multi-scale ST correlations in the MSST block.

### B. Graph Attention for Extracting Spatial Features

The spatial extractor uses the Graph Attention mechanism to learn the degree of spatial correlations $\mathbf{A}(t) = [a_{ij}]$ $(0 < a_{ij}(t) < 1)$, which captures the spatial dynamic. The input of the spatial extractor is defined as $\mathbf{S}^{-}_{in}(t;T_s) = (\mathbf{s}^{-}_{in,1}(t;T_s), \ldots, \mathbf{s}^{-}_{in,N}(t;T_s)) \in \mathbb{R}^{N \times CS_{in} \times T_s}$, where $\mathbf{s}^{-}_{in,i}(t;T_s) \in \mathbb{R}^{CS_{in} \times T_s}$ is the input data of node $i$ $(1 \le i \le N)$; $CS_{in}$ is the number of input channels and $T_s$ is the length of the time dimension; The outputs $\mathbf{S}^{-}_{out}(t;T_s) \in \mathbb{R}^{N \times CS_{out} \times T_S}$ ($CS_{out}$ is the number of output channels) is generated as follows.

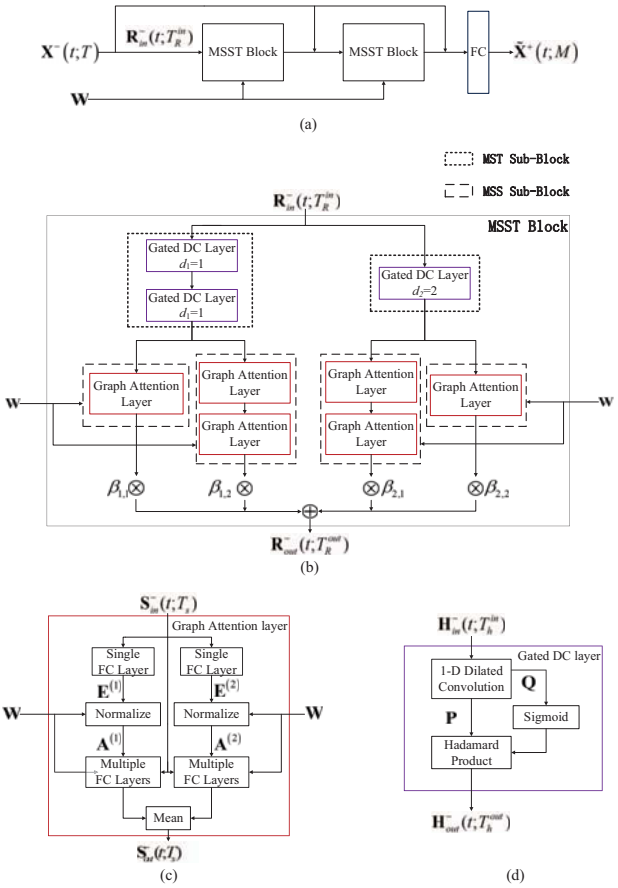



Fig. 2: (a) Architecture of MSTNN$(2,2)$. (b) Structure of the MSST block. The MSST block are consisted of two kinds of extractors, the spatial extractor Graph Attention layer and the temporal extractor Gated DC layer. Each of the final four branches represents a scale of ST correlation. (c) Structure of a Graph Attention layer with $K = 2$ ($K$ is the number of branches to independently perform graph attention mechanism). (d) Structure of a Gated DC layer.

Step 1: Transform the input into higher-level features using a linear transformation with weights $\mathbf{\Theta}_1 \in \mathbb{R}^{CS_{out} \times CS_{in}}$) and then apply the self-attention mechanism on each pair of nodes to obtain attention coefficients,

$$e_{ij}(t) = f(\mathbf{\Theta}_1 \mathbf{s}^{-}_{in,i}(t;T_s), \mathbf{\Theta}_1 \mathbf{s}^{-}_{in,j}(t;T_s)). \tag{4}$$

where $j \in \mathcal{N}_i = \{j | W_{ij} = 1, j \in 1,2,\ldots,N\}$. In this paper, the self-attention mechanism $f$ is implemented by a single-layer feed forward neural network, using LeakyReLU as the activation function.

Step 2: Normalize $e_{ij}(t)$ across all choices of $j$,

$$a_{ij}(t) = \text{softmax}(e_{ij}(t)) = \frac{\exp(e_{ij}(t))}{\sum_{v \in \mathcal{N}_i} \exp(e_{iv}(t))}. \tag{5}$$

Step 3: Apply nonlinearity on a linear combination of the features corresponding to $a_{ij}(t)$ to attain the final output features of every node,

$$\mathbf{s}^{-}_{out,i}(t;T_s) = \sigma\Big(\sum_{j \in \mathcal{N}_i} a_{ij}(t) \mathbf{\Theta}_2 \mathbf{s}^{-}_{in,i}(t;T_s)\Big), \tag{6}$$

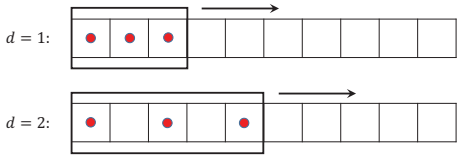



Fig. 3: Dilated convolution with dilated factor 1 and 2 respectively. The red dots are locations of kernel elements (The kernel size is 3).

where $\mathbf{\Theta}_2 \in \mathbb{R}^{CS_{out} \times CS_{in}}$ and $\sigma$ is an activation function, i.e., Sigmoid function in this paper. Compared with Eq. 1 and 2, Eq. 6 is able to catch the time-varying spatial correlations by learning $a_{ij}(t)$ from historical data.

Step 4: In order to stabilize the learning process of the self-attention mechanism, the features of $K$ times independent attention procedures is averaged by using multi-head attention technique [17],

$$\mathbf{s}^{-}_{out,i}(t;T_s) = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} a^k_{ij}(t)\mathbf{\Theta}^k_2 \mathbf{s}^{-}_{in,j}(t;T_s)\right), \quad (7)$$

where $\mathbf{\Theta}^k_2$ and the weights in $a^k_{ij}(t)$ are initialized independently for every $k$ branch. Particularly, $K = 2$ in Fig. 2(d).

To be noticed, the features within the $l$-hop neighbors can be extracted by stacking $l$ Graph Attention layers.

### C. Gated Dilated Convolution for Extracting Temporal Features

The temporal extractor is built with the enhanced Gated CNN. The basic Gated CNN performs 1-dimension standard convolutions sequentially,

$$(\Gamma * y)(p) = \sum_{s+l=p} \Gamma(l)\, y(s), \quad (8)$$

where $y : \mathbb{Z} \to \mathbb{R}$ is a discrete function; $\Omega_r = [-r, r] \cap \mathbb{Z}$ and $\Gamma : \Omega_r \to \mathbb{R}$ is a kernel with size $(2r+1)$. As using a consecutive kernel $\Gamma$, the basic Gated CNN misses temporal information in non-adjacent time steps.

To escape this dilemma, we design a gated dilated convolution (Gated DC) layer by replacing the standard convolution in Gated CNN with the dilated convolution [18], see Fig. 2(c). The dilated convolution $*^{(d)}$ with a dilation factor $d$ is expressed as,

$$\left(\Gamma *^{(d)} y\right)(p) = \sum_{s+dl=p} \Gamma(l)\, y(s). \quad (9)$$

In this way, dilation factors $d$ acts as the granularity of temporal correlations. The 1-dimensional dilated convolutions with $d = 1$ and $d = 2$ are shown in Fig. 3. Notice that the dilated convolution with dilation factor $d$ shortens the length of a sequence by $d(k-1)$ ($k = |\Omega_r|$ is the kernel size). Therefore, the lengths of the outputs in Gated DC layers with different $d$ are not the same.

The procedure at the Gated DC layer is as follows. Let $\mathbf{H}^{-}_{in}(t;T^{in}_h) = (\mathbf{h}^{-}_{in,1}(t;T_h),\ldots,\mathbf{h}^{-}_{in,N}(t;T^{in}_h)) \in \mathbb{R}^{N\times CT_{in}\times T^{in}_h}$ denote the input of the Gated DC layer ($CT_{in}$ is the input channels and $T^{in}_h$ is the length in the input time dimension). The output of the Gated DC layer is defined as $\mathbf{H}^{-}_{out}(t;T^{out}_h) = (\mathbf{h}^{-}_{out,1}(t;T^{out}_h),\ldots,\mathbf{h}^{-}_{out,N}(t;T^{out}_h)) \in \mathbb{R}^{N\times CT_{out}\times T^{out}_h}$ ($CT_{out}$ is the input channels and $T^{out}_h$ is the length of the input time dimension), where

$$\mathbf{h}^{-}_{out,i}(t;T^{out}_h) = \mathbf{P}_i \odot \sigma(\mathbf{Q}_i) \in \mathbb{R}^{CT_{in}\times T^{out}_h}, i = 1,\ldots,N, \quad (10)$$

where $\mathbf{P}_i = \mathbf{\Gamma}^p_i *^{(d)} \mathbf{h}^{-}_{in,i}(t;T^{in}_h)$, $\mathbf{Q}_i = \mathbf{\Gamma}^q_i *^{(d)} \mathbf{h}^{-}_{in,i}(t;T^{in}_h)$, $\mathbf{\Gamma}^p_i, \mathbf{\Gamma}^q_i \in \mathbb{R}^{k\times CT_{in}\times CT_{out}}$ (the $*^{(d)}$ here means perform the 1-d dilated convolution in Eq. 9 along the rows of $\mathbf{h}^{-}_{out,i}(t;T^{out}_h)$ independently); $\odot$ is the Hadamard product; $T^{out}_h = T^{in}_h - d(k-1)$ is the shortened output length.

### D. Fusing Multi-scale ST correlations

To make our MSTNN more powerful, we design the MSST block to explore and integrate multi-scale ST correlations. Recall that the MSST block is composed of $n$ MST sub-blocks and $mn$ MST blocks. The Gated DC layers in each MST sub-block have the same dilation factor and extract the temporal correlations from specific time steps. Therefore, the temporal scales of the MSTNN is $n$. On the other hand, the $mn$ MST blocks can be classified into $m$ classes, each of which contains the same number of Graph Attention layers to extract the spatial correlations from specific neighbors with the same hop. Therefore, the spatial scales of the MSTNN is $m$. The MSTNN with spatial scale $m$ and temporal scale $n$ is then denoted by MSTNN$(m, n)$.

It is to be noticed that the number of Gated DC layers should satisfy the following condition. Suppose the input length of the MST sub-block is $T_{in}$ and output length is $T_{out}$. The $j$-th MST sub-block with dilation factor $d_j$ and kernel size $k_j$ should be composed of $l_j$ dilated convolution layers, where

$$l_j = \frac{T_{in} - T_{out}}{d_j(k_j - 1)}, \ 1 \le j \le n. \quad (11)$$

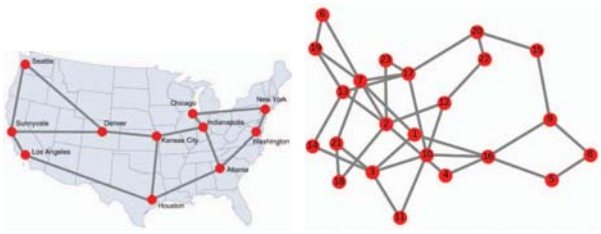
Note that $d_j$ and $k_j$ are set in your own manner.

By cascading MST and MSS sub-blocks, we can get ST correlations with different spatial and temporal scales. It is necessary to fuse ST correlations of all scales to generate the outputs of the MSST block,. Benefited from the settings to assimilating output sizes of different branches, we adopt the weighted sum fusing method, which can separate the importance of ST correlations with different scales.

To fuse the spatial and temporal correlations, we set the structure of MSTNN$(m, n)$ as shown in Fig. 2 (b). The ST correlation weight is then $\mathbf{B} \in \mathbb{R}^{n\times m}$, where $\beta_{i,j}$ is the importance weight of the ST correlation generated by MST sub-block $i$ and MSS sub-block $j$. To keep the ST correlations reliable, $\mathbf{B}$ should satisfy $\sum_{i,j}\beta_{i,j} = 1$.

## IV. Simulations

In this section, we test the prediction performance of MSTNN$(1, 1)$, MSTNN$(1, 2)$, and MSTNN$(2, 2)$ on real network traffic data. We use the Abilene dataset [13] and the GÉANT dataset [14], which provide public intra-domain



(a) (b)

Fig. 4: Network topology. (a) The Abilene network [13]. (b) The GÉANT network [14].

traffic matrices in the IP backbone networks of the USA and Europe respectively. To show the effectiveness of MSTNN for solving the ODTP problem in terms of prediction accuracy, we compare our three MSTNNs with existing baselines on the above two datasets.

### A. Dataset Description

The Abilene dataset, which is collected from the Abilene network, consists of 11 nodes, 14 links, and 121 OD flows. The topology of the Abilene network is shown in Fig. 4(a). The data are collected continuously over one week and contain 2016 consecutive 5-minute intervals over the week.

The GÉANT dataset is collected from the pan-European research and education network GÉANT, which is composed of 23 routers, 38 links, and 529 OD flows. The topology of GÉANT is shown in Fig. 4(b). The data are collected continuously every 15 minutes over four months, that is, 10772 records in total.

### B. Experimental Settings

All experiments are tested on a Linux server (GPU: NVIDIA GeForce GTX 1080Ti). 30 historical observed data points ($T = 30$) are used to forecast traffic volume in the next adjacent three time steps ($M = 3$). In the Abilene dataset, $M = 3$ means to predict OD traffic volume in the next 5, 10, and 15 minutes. While in the GÉANT dataset, $M = 3$ is to forecast future OD traffic volume in the next 15, 30, and 45 minutes. The numbers of input channels in the MST sub-block and MSS sub-block are set to 64 and 16, respectively. It is to be noticed that the number of input channels of the MST sub-blocks in the first MSST block is $N$. Data in both datasets are normalized by the Z-Score method [9].

*1) Evaluation Metrics:* To evaluate the performance of different models, Mean Absolute Errors (MAE), Mean Squared Errors (MSE), and Mean Absolute Percentage Errors (MAPE) are adopted.

*2) Our Models:* Besides MSTNN$(2, 2)$, MSTNN$(1, 1)$ and MSTNN$(1, 2)$ are also in comparison, to separately show the effectiveness of the graph attention mechanism based spatial correlation extractor, the dilated convolution based temporal correlation extractor and multi-scale ST correlations.

TABLE I: Characteristics of Different Models

| | SFP | MFP | ML [1] | GL [2] | TVS [3] | LTT [4] | MSST [5] |
|---|---|---|---|---|---|---|---|
| HA | √ | × | × | × | × | × | × |
| ARIMA | √ | × | × | × | × | × | × |
| CNN | × | √ | √ | × | × | × | × |
| FC-LSTM | √ | × | √ | × | × | × | × |
| STGCN | × | √ | √ | √ | × | × | × |
| MSTNN$(1, 1)$ | × | √ | √ | √ | √ | √ | × |
| MSTNN$(1, 2)$ | × | √ | √ | √ | √ | √ | √ |
| MSTNN$(2, 2)$ | × | √ | √ | √ | √ | √ | √ |

[1] Machine Learning, [2] Graph Learning, [3] Time-varying Spatial Correlations, [4] Long-term Temporal Correlations, [5] Multi-scale Spatial-Temporal Correlations. For example, HA is a statistical method to solve single flow prediction problem.

*3) Baselines:* a) Historical Average (HA); b) Auto-Regressive Integrated Moving Average (ARIMA) [5]; c) Convolution neural network (CNN) [6]; d) Full-Connected LSTM (FC-LSTM) [19]; e) Spatio-temporal Graph Convolutional Network (STGCN) [9]. The characteristics of baselines and our models are shown in Table I.

### C. Experiment Results

Table II and Table III describe the performance of different prediction methods on the datasets Abilene and GÉANT, respectively. It can be seen that MSTNN consistently achieves the best performance in terms of all evaluation metrics. We can also observe that machine learning methods generally outperform statistical methods ARIMA and HA.

*1) Benifits of Topology Information:* Comparing the results on the two datasets in Table II and Table III, all graph-learning based methods perform better than grid-structure-based methods, illustrating that the ST correlations are related to the network topology. The advantage of graph-learning based methods, STGCN and MSTNN, on the GÉANT dataset is more obvious than on the Abilene dataset, since the network

TABLE II: Performance comparison on the Abilene network

| Model | Abilene(5/10/15 min) | | |
|---|---|---|---|
| | MAE($\times 10^3$) | MSE($\times 10^3$) | MAPE(%) |
| HA | 8.25 | 14.07 | 14.24 |
| ARIMA | 9.54/9.93/10.85 | 16.25/16.84/17.95 | 15.05/16.11/17.23 |
| CNN | 7.26/8.29/9.34 | 12.99/14.85/16.23 | 11.26/12.83/14.01 |
| FC-LSTM | 6.12/7.41/8.50 | 12.03/13.64/15.18 | 10.95/12.44/13.70 |
| STGCN | 5.43/6.26/7.45 | 10.82/12.87/14.39 | 9.38/10.89/13.25 |
| **MSTNN$(1, 1)$** | **5.36/6.14/7.21** | **10.66/12.49/14.04** | **9.29/10.80/13.05** |
| **MSTNN$(1, 2)$** | **5.24/6.11/7.18** | **10.54/12.21/13.89** | **9.18/10.72/12.91** |
| **MSTNN$(2, 2)$** | **5.15/5.96/7.01** | **10.17/11.77/12.95** | **9.02/10.45/12.78** |

TABLE III: Performance comparison on the GÉANT network

| Model | GÉANT(15/30/45 min) | | |
|---|---|---|---|
| | MAE($\times 10^3$) | MSE($\times 10^3$) | MAPE(%) |
| HA | 14.63 | 21.57 | 9.97 |
| ARIMA | 16.96/18.41/20.05 | 24.55/30.10/35.43 | 11.51/12.58/13.06 |
| CNN | 12.21/16.32/18.97 | 18.28/24.93/33.07 | 8.36/10.55/11.94 |
| FC-LSTM | 11.35/14.05/17.21 | 15.41/22.86/31.52 | 7.92/10.34/11.35 |
| STGCN | 8.49/11.06/12.99 | 12.36/17.78/27.44 | 5.86/8.02/9.71 |
| **MSTNN$(1, 1)$** | **8.23/10.83/12.63** | **11.93/17.42/26.47** | **5.83/7.97/9.63** |
| **MSTNN$(1, 2)$** | **8.01/10.65/12.30** | **11.72/15.47/25.59** | **5.77/7.56/9.40** |
| **MSTNN$(2, 2)$** | **7.89/10.22/12.11** | **11.41/15.04/24.97** | **5.53/7.48/9.21** |

of the GÉANT network is more complicated, as shown in Fig. 4. The graph-learning based methods can effectively utilize spatial structure to make more accurate predictions.

*2) Benefits of Graph Attention:* To illustrate the advantage of the graph attention mechanism, we compare STGCN and MSTNN$(1,1)$. Because MSTNN$(1,1)$ does not consider multi-scale temporal correlations, the only difference between STGCN and MSTNN$(1,1)$ lies in the spatial correlation extractor. STGCN uses the spectral graph convolution in Eq. 2 that presets the spatial correlations, while MSTNN employs the graph attention mechanism in Eq. 7 to adjust the extraction of spatial correlations with the change of inputs. It is easily seen that MSTNN$(1,1)$ outperforms STGCN in all three metrics, indicating that the graph attention mechanism is useful in capturing time-varying spatial correlations.

*3) Benefits of Multi-Scale Temporal Correlations:* The benefits of extracting the multi-scale temporal correlations are shown clearly by comparing MSTNN$(1,1)$ with MSTNN$(1,2)$. The only difference between MSTNN$(1,1)$ and MSTNN$(1,2)$ is the temporal scales. MSTNN$(1,2)$ performs better than MSTNN$(1,1)$ in terms of all three evaluation metrics, illustrating the necessity of multi-scale temporal correlations. In addition, because dilation factors larger than 1 equivalently enlarge the granularity of temporal correlations, the better performance of multi-scale temporal correlation also validates the necessity to utilize the long-term trend in OD traffic data.

*4) Benefits of Multi-Scale Spatial Correlations:* The comparison between MSTNN$(2,2)$ and MSTNN$(1,2)$ explicitly demonstrates the effectiveness of multi-scale Spatial correlations. The only difference between them is that MSTNN$(2,2)$ considers both 1 and 2-hop neighbors, while MSTNN$(1,2)$ only considers 1-hop neighbors. The results that MSTNN$(2,2)$ outperforms MSTNN$(1,2)$ reveal that the 2-hop neighbors are helpful in the OD traffic prediction. This also validates our deduction about the effectiveness of long-term trend in OD traffic, because the OD traffic may reach the 2-hop neighbors after a few time steps.

*5) Benefits of Multi-Scale ST Correlations:* By comparing the performance of MSTNN$(2,2)$ with MSTNN$(1,2)$ and MSTNN$(1,1)$, it can be seen that MSTNN$(2,2)$ achieves the best prediction accuracy consistently, which validates the advantage of multi-scale ST correlations. It is to be noticed that we only validate the effectiveness of multi-scale ST correlations, and we will study the problem of the best spatial and temporal scales in our future work.

## V. Conclusions

In this paper, we constructed the OD traffic prediction problem on graphs and proposed a novel graph learning method, MSTNN. MSTNN is capable of handling graph-structured time-series and making predictions. Experiments showed that MSTNN outperformed existing approaches on two real-world datasets, indicating its great potentials in solving the graph-based ODTP problem. Compared to the methods with a single scale, MSTNN was more powerful by extracting multi-scale ST correlations. In the future, we will extend MSTNN to other time-varying networks, e.g., wireless sensor networks, and apply it to traffic management.

## VI. Acknowledgement

This work was supported by the Fundamental Research Funds for the Central Universities (ZYGX2019J121).

## References

[1] L. Khoukhi and S. Cherkaoui, "Experimenting with fuzzy logic for qos management in mobile ad hoc networks," *International Journal of Computer Science and Network Security*, vol. 8, pp. 372–386, 09 2008.

[2] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, A. Nucci, A. Nucci, M. Crovella, and C. Diot, "Traffic matrices: balancing measurements, inference and modeling," in *ACM SIGMETRICS*, 2004.

[3] L. Khoukhi and S. Cherkaoui, "A quality of service approach based on neural networks for mobile ad hoc networks," in *Second IFIP International Conference on Wireless and Optical Communications Networks, 2005. WOCN 2005.*, 2005.

[4] M. Mardani and G. B. Giannakis, "Estimating traffic and anomaly maps via network tomography," *IEEE Transactions on Networking*, vol. 24, no. 3, pp. 1533–1547, 2016.

[5] G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[6] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818–834, 2017.

[7] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, July 2017.

[8] Y. Li, R. Yu, C. Shahabi, and L. Yan, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proceedings of International Conference on Learning Representations*, 2018.

[9] Y. Bing, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of International Conference on Learning Representations*, 2017.

[11] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proceedings of International Conference on Learning Representations*, 2017.

[13] Yin Zhang's Abilene TM. [Online]. https://www.cs.utexas.edu/users/yzhang/research/AbileneTM/.

[14] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *Acm Sigcomm Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[15] D. Duvenaud, D. Maclaurin, J. Aguileraiparraguirre, R. Gómezbombarelli, T. Hirzel, A. Aspuruguzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in Neural Information Processing Systems*, 2015, pp. 2224–2232.

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

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[18] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of International Conference on Learning Representations*, 2016.

[19] X. Shi, Z. Chen, W. Hao, D. Y. Yeung, W. Wong, and W. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.