



# Deep learning based origin-destination prediction via contextual information fusion

Hao Miao<sup>1</sup> · Yan Fei<sup>2</sup> · Senzhang Wang<sup>1,3</sup> · Fang Wang<sup>4</sup> · Danyan Wen<sup>5</sup>

Received: 29 July 2020 / Revised: 20 November 2020 / Accepted: 29 December 2020 /

Published online: 30 January 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

## Abstract

Origin-Destination (OD) prediction which aims to predict the number of passenger's travel demands from one region to another, is critically important to many real applications including intelligent transportation systems and public safety. The challenges of this problem lie in both the dynamic patterns of the human mobility data and data sparsity in issue in some regions. Thus it is difficult to model the complex spatio-temporal correlations of the human mobility data to predict the OD of their trips. Meanwhile, the crowd flows in different regions of a city and the context features (e.g. holiday, weather and POIs) are potentially useful to alleviate the data sparsity issue and improve the OD prediction, but are largely ignored by existing works. In this paper, we propose a deep spatio-temporal framework which named Auxiliary-tasks Enhanced Spatio-Temporal Network (AEST) to more effectively address the OD prediction problem. AEST trains a model to conduct OD inference via learning crowd flow and external data as auxiliary task. The novel Hierarchical Convolutional LSTM (HC-LSTM) Network is proposed which combines CNN, GCN and LSTM to effectively capture spatio-temporal correlations. In addition, we design a Contextual Network (ContextNet) which learns representations of contextual information to assist OD prediction. We conduct extensive experiments over bike and taxicab trip datasets in New York. The results show that our method is superior to the state-of-art approaches.

**Keywords** OD prediction · Spatio-Temporal network · Auxiliary task

## 1 Introduction

Recently, ride-hailing applications, such as Didi, Uber and UCAR, have experienced a tremendous expansion as it brings great convenience to ride service and improves the efficiency of public transportation. Traffic prediction is one of the most popular research problem of spatio-temporal prediction. Most of existing work only focus on predicting

---

✉ Senzhang Wang  
szwang@nuaa.edu.cn

inflow and outflow, which is shown in Fig. 1a, in all regions or some specific locations, while ignoring the influence of Origin-Destination(OD) demands. The illustration of OD demands is shown in Fig. 1b which shows the number of passenger's travel demands from one geographical region to another in a given time slot. Taking region  $r_1$  to  $r_4$  as an example, we can see the OD demand from  $r_3$  to  $r_1$  is 1, and the OD demand from  $r_1$  to  $r_4$  is 1. In this paper, we investigate the problem of OD prediction with the help of crowd flow data and external information. Estimating OD demands is of great importance to various practical ride-hailing applications, and attracts rising research attention recently. To provide high-quality services and achieve company profits, ride-hailing platforms need to fully understand the passenger travel demands in real time. On one hand, the ride-hailing platforms must pre-assign service vehicle in advance so as to satisfy passenger demands. On the other hand, it is crucial to maximize the profit throughout understanding the underlying travel regularities from historical passenger demands, thus avoiding driving without passengers. In addition, the passenger's pick-up/drop-off demand is especially helpful for emerging mobility-on-demand(MOD) services in terms of more efficient vehicle distribution.

Due to the importance of this problem, a lot of efforts have been made to addressing it. [16] proposed a Contextualized spatial-temporal network(CSTN) for taxi OD demand prediction which combines CNN and LSTM to model spatio-temporal dependency. Wang et al. [23] proposed deep learning based model named GEML that employs GCN [11] and Peridic-Skip LSTM, to forecast OD demands. To consider the relations between a pair of regions, [31] proposed a multi-task framework MDL to predict node flow and edge flow simultaneously. Although these works try to combine crowd flow prediction and flow OD prediction by considering the high correlation of the two tasks, how crowd flow can be used to facilitate OD prediction is not well studied. Moreover, the complex spatial and temporal correlations are not well captured by existing works, either.

In this paper, we study the novel problem of OD prediction via contextual information fusion. Our insight is that contextual information like inflow/outflow and external features(i.e., weather) are complementary to OD prediction. First, previous works

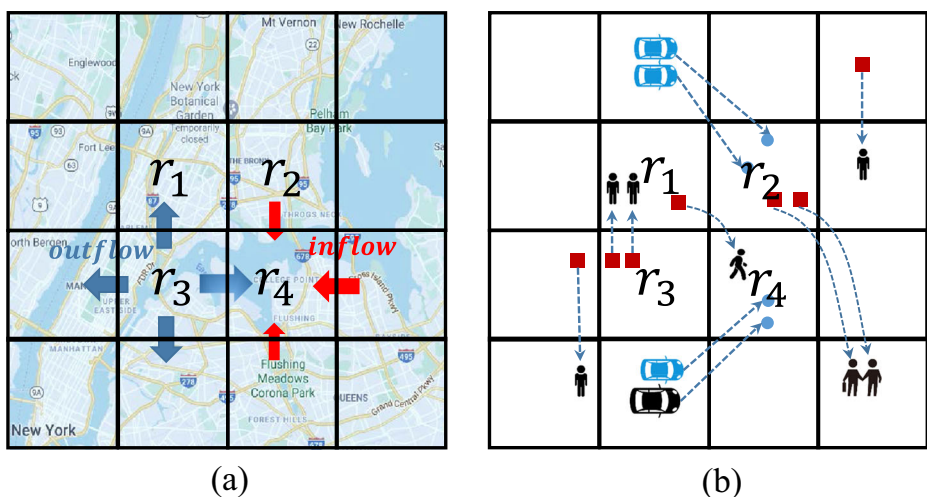


Fig. 1 Illustration of In/Out flow and OD

demonstrated that crowd flow is helpful for spatial-temporal prediction [22, 28, 31] including OD prediction. Second, as shown in Fig. 1, the left part shows crowd flows including in- and out- flows reflecting the human mobility dynamics in different areas of a city, while the right part shows where the flows are from indicating the origin-destination of the flows. The inflows of a region can be obtained by merging all the OD flows whose destination is in this region, while the outflows of a region can be obtained by merging all the OD flows whose origin is in this region. It is clear that crowd flows are highly correlated to OD of the flows.

However, this problem is non-trivial to address due to the following challenges. First, it is difficult to mine the correlations between OD demands and contextual information effectively to facilitate OD prediction. Although some multi-task model are proposed [31], how to use contextual information to assist OD prediction is not well studied. Second, the spatial and temporal correlations of the OD demands are complex, and thus cannot be easily captured. Recently, some deep learning based OD prediction models try to employ CNN to capture spatial correlations. However, CNN can only capture geographical similarity, but ignore the influence of semantic correlation [25] that two locations could be spatially distant but are similar in their demand patterns. Although [23] propose GEML to capture semantic correlation by graph convolutional network [11]. It is still a non-trivial to effectively combine the geographical relevance and semantic correlation. As the human mobility changes over time, it is even more difficult to capture the regularity of human mobility patterns. Third, data sparsity issue is common in OD demand data (e.g., ride-hailing records). There might be thousands of demands in the downtown, while very few demands in the suburbs.

To address the aforementioned challenges, we propose a novel Auxiliary-tasks Enhanced Spatio-Temporal Network (AEST). AEST is a Seq2Seq based hierarchical spatio-temporal network which first fuses GCN and CNN to capture spatial dependencies in terms of geographical and semantic correlations, and then input the representations into LSTM to learn temporal representations. In the data preprocessing stage, we convert original trajectories data to image like data (e.g., Crowd flow image and OD image). The data representation of crowd flow image and OD image are not effective to explicitly reflect the semantic spatial correlations, due to the image like data may not follow the spatial smoothness property. To capture the global features, we construct semantic graph. Then, To well model spatial and temporal dependencies, we propose a hierarchical Convolutional LSTM (HC-LSTM) Network to extract the OD representation. In our network, two auxiliary networks are proposed to extract crowd flow and external features separately. Then we fuse the two types of features to get a unified auxiliary-task representation. We incorporate this auxiliary-task representation into seq2seq model which deeply captures the relationships between crowd flow and OD data to assist OD prediction. In this way, the OD data sparsity issue can be alleviated by adding knowledge of crowd flow data and external information. We evaluate the proposed method on large-scale real-world public datasets including taxi data of New York city (NYC) and bike-sharing data of NYC. Experiment results show the competitive superiority of the proposed model AEST by comparing with the state-of-the-art methods.

Our major contributions are summarized as follows.

- We propose a Auxiliary-tasks Enhanced Spatio-Temporal Network (AEST) which can effectively integrate the crowd flow features and external context features to improve OD prediction.
- We propose a Seq2Seq based hierarchical spatio-temporal network to model the complex spatial similarity and dynamic temporal dependency in a unified way.
- We conduct experiments on several real world traffic datasets. The results show that our model is consistently better than other state-of-the-art methods.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. Section 3 outlines the preliminary concepts and formulates the problem. Section 4 details the structure of the proposed model. Section 5 describes the experimental results. Finally, Section 6 concludes the paper and discuss the future application.

## 2 Related work

**Traffic prediction** Traffic prediction becomes more and more popular due to the increasingly available urban data (e.g., *taxi trajectories*) and rich applications (e.g., *Uber*). Traditionally, statistic-based methods such as ARIMA and SVR, are used as traffic flow prediction model. Cetin and Comert [3] put forward a regression model which included two kinds of traffic incident detection algorithms for traffic flow prediction. ARIMA [13, 24] is used to predict the short-term traffic flow. Some work improve the original ARIMA [4] to study the change rules of the traffic flow, and the tuning proportion matrix is introduced to improve the prediction accuracy of the short-term traffic flows. However, statistic-based model does not have the ability to model the complex spatial and temporal correlations due to their limited learning ability.

**Crowd flow prediction** Crowd flow prediction is a typical spatio-temporal data prediction task which focuses on predicting the traffic over cell regions. In recent years, with the advances of deep learning techniques, deep neural network based models [7, 26, 32] are widely used in crowd flow prediction. A common practice in most existing work is treating entire city as images, and dividing the city into small regions which is similar to pixel in image according to latitude and longitude, so that CNN [12] can be applied. In addition, RNN like LSTM [9] is used to capture temporal correlation. Some studies treat the traffic flow data of the entire city as images, and then applied CNN to model the spatial correlations. Zhang et al. [29] proposed a CNN based model STResNet to forecast inflow and outflow in each cell region of a city. Other studies combined CNN and RNN to model spatial and temporal dependency simultaneously. Shi et al. [19] proposed a Convolutional LSTM (ConvLSTM) Network to predict precipitation. Wang et al. [20] proposed a Seq2Seq framework which named SeqST-GAN, which applied GAN and attention mechanism, to predict multi-step crowd flows.

A convolutional neural network (CNN) is able to exploit the shift-invariance, local connectivity, and compositionality of image data. However, traditional deep learning, like CNNs, can just capture hidden state of Euclidean data. In real world, Non-Euclidean are ubiquitous, especially in the form of graphs. So there is an increasing interest in extending deep learning approaches for graph data which is called graph neural networks (GNNs). Li et al. [14] proposed Diffusion Convolutional Recurrent Neural Network (DCRNN) to predict traffic flow in a graph manner. Yu et al. [27] propose a novel deep learning framework, Spatio-Temporal Graph Convolutional Networks (STGCN), to tackle the time series prediction problem in traffic domain. Diao et al. [6] proposed a dynamic spatio-temporal GCNN for accurate traffic forecasting. In addition, [21] provided a comprehensive survey on deep learning based spatio-temporal data mining methods and applications. Lin et al. [15] proposed a deep learning based convolutional model to predict crowd flows in the metropolis. However, these works only focus on crowd flow prediction, but ignore the Origin-Destination prediction.

**OD Prediction** Origin-Destination(OD) prediction which aims to predict the number of passenger demands from one region to another, is beneficial to many real applications such

as traffic management and ride-hailing services. Traditional methods [1, 2] mostly used regression based approaches or other statistics-based approaches to predict or estimate the dynamic vehicle OD matrix in a transportation network. Recently, [17] modeled the temporal OD trip matrix as a four-order tensor consisting of four attributes: origin, destination, vehicle type and time, and then proposed to use tensor decomposition technique to forecast future traffic demand. Wang et al. [23] proposed a Grid-Embedding based Multi-task learning (GEML) which applied GCN and LSTM modeling spatio-temporal dependency simultaneously, to predict OD matrix and crowd flow. Liu et al. [16] proposed a Contextualized Spatio-Temporal Network (CSTN) to predict the taxi demand between all region pairs in future time interval. Chu et al. [5] developed a deep learning model called multi-scale convolutional long short-term memory network (Multi-ConvLSTM) to predict the future travel demand and the OD flows. Zhang et al. [31] proposed a multi-task deep learning framework to predict the flow and OD simultaneously throughout a spatio-temporal network MDL. Zhang et al. [30] proposed an indicator called OD attraction degree (ODAD) for OD prediction.

However, most existing works consider OD prediction and crowd flow prediction as two separate tasks while ignoring the high correlations between them. Although [31] proposed a multi-task learning framework MDL to predict flow and OD at the same time. It just simply concatenates the features of crowd flow and OD. How to effectively use the knowledge of flow to assist OD prediction still remains an open problem.

### 3 Notations and problem definition

In this section, we will first give some notations to help us state the studied problem. Next, a formal problem definition will be given.

**Definition 1 Cell Region** The city under study is divided into a  $C = m \times n$  grid map based on the latitude and longitude. Each grid represents an equal-sized cell region. We denote all the cell regions as  $R = \{r_{1,1}, \dots, r_{i,j}, \dots, r_{m,n}\}$ , where  $r_{i,j}$  represents the  $i$ -th row and  $j$ -th column cell region of the grid map.

**Definition 2 Flow Image** Let  $\mathcal{P}$  be a collection of crowd flow trajectories. Given a cell region  $r_{i,j}$ , the corresponding inflow and outflow of the crowds in time slot  $t$  can be defined as

$$\begin{aligned} x_{in,i,j}^t &= \sum_{T_r \in \mathcal{P}} |\{l > 1 | g_{l-1} \notin r_{i,j} \wedge g_l \in r_{i,j}\}| \\ x_{out,i,j}^t &= \sum_{T_r \in \mathcal{P}} |\{l > 1 | g_l \in r_{i,j} \wedge g_{l+1} \notin r_{i,j}\}| \end{aligned} \quad (1)$$

where  $T_r : g_1 \rightarrow g_2 \rightarrow \dots \rightarrow g_{T_r}$  is a trajectory at time slot  $t$  in  $\mathcal{P}$ , and  $g_l$  is the geospatial coordinate;  $g_l \in r_{i,j}$  means  $g_l$  is within region  $r_{i,j}$ ;  $|\cdot|$  denotes the cardinality of a set. The illustration of inflow and outflow is shown as Fig. 1[a]. Following [29], we denote the inflow and outflow of all the cell regions in  $t$  as a crowd flow tensor  $\mathcal{X}^t \in \mathcal{R}^{m \times n \times 2}$ , where  $\mathcal{X}_{i,j,0}^t = x_{in,i,j}^t$ ,  $\mathcal{X}_{i,j,1}^t = x_{out,i,j}^t$ .

**Definition 3 OD Image** We define OD demands at time slot  $t$  as a matrix  $D^t \in \mathcal{R}^{N \times N}$ , where  $N = m \times n$  is the number of regions and each elements  $d_{i,j}^t$  denotes the sizes of

flows starting from  $i$ -th cell region and ending at  $j$ -th cell region of  $R$  as illustrated in the Fig. 1b. Based on the OD matrix  $D^t$ , we denote the OD demands of all the cell regions named OD Image in time-slot  $t$  as a OD tensor where  $\mathcal{M}^t \in \mathcal{R}^{m \times n \times N}$  with  $N$  channels,  $\mathcal{M}_{i,j,n}^t$  denotes the number of OD demands from the  $n$ -th region to region  $r_{i,j}$ .

**Definition 4 Semantic Graph** We define the semantic spatial-temporal graph at time slot  $t$  as  $\mathcal{G}^t = \{V, E^t\}$ , whose nodes  $V$  are the cell regions. There is an edge  $e_{i,j}^t$  if there are flow trajectories whose origin is  $v_i$  and destination is  $v_j$ . Note that the weight  $w_{i,j}^t$  is set to 1.

**Problem 1** Given the OD images  $\{\mathcal{M}^t, \mathcal{G}^t | t = 1, \dots, T\}$  in the cell regions  $R$  over  $T$  time-slots, the flow images  $\{\mathcal{X}^t | t = 1, \dots, T\}$  and the external information data  $E$  (e.g., weather, holiday, etc.), our goal is to predict the OD image  $\mathcal{M}^{t+1}$  in the future.

## 4 Methodology

### 4.1 Model framework

The overall architecture of our model AEST is illustrated in Fig. 2. The model input consists of two parts, the OD image  $\mathcal{M}$  and semantic graph  $\mathcal{G}$ . *OD Encoder* first extracts the spatio-temporal features of OD data using *HC-LSTM*. The residual connection [8] is used in *OD Encoder* to avoid overfitting. Then crowd flow images and external features are fed into Contextual Network to learn the latent features respectively. Next, the crowd flow features and external features are fused with a concatenation operation, which can be represented as contextual features. Finally, we combine OD features and contextual features, and feed it into *OD Decoder* to predict future OD demands.

### 4.2 Data preprocessing

Based on Definition 2-4, given the crowd flow trajectories  $\mathcal{P}$ , we first need to convert them to three types of data forms, flow images, OD images, and Semantic graphs. Following the

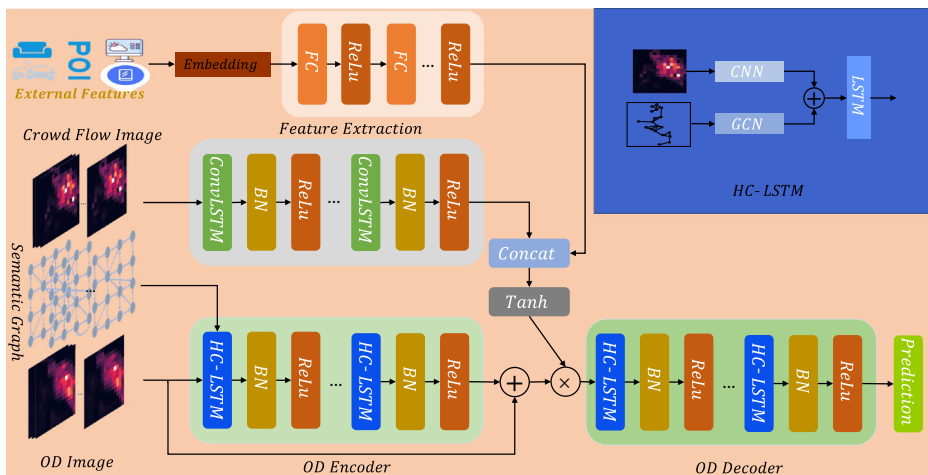
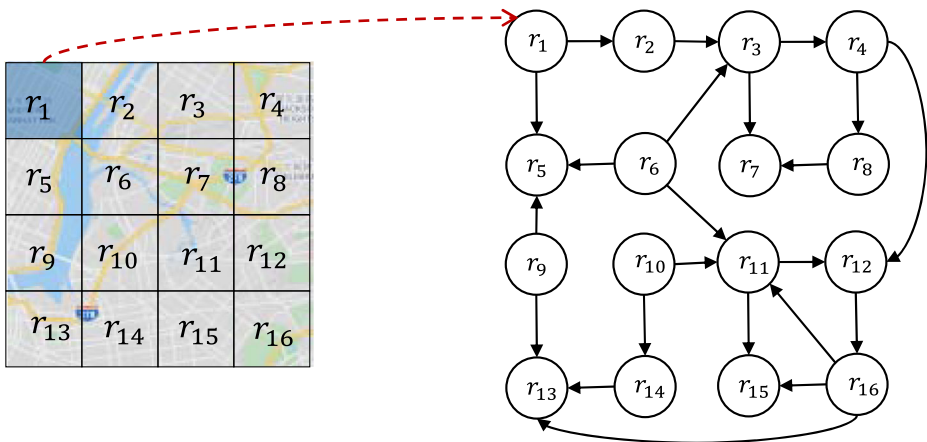


Fig. 2 The framework of our model



**Fig. 3** Illustration of graph construction

previous work [29], we first model the crowd flow images with the size of  $m \times n \times 2$  as time-varying spatial maps which can be represented as time-ordered sequence of tensors, so that convolution operation can be applied for feature learning. similarly, we construct OD image based on origin and destination of the raw trajectories with the size of  $m \times n \times C$ , where  $C = m \times n$ . The illustration of semantic graph construction is shown in Fig. 3. To capture the global features, we construct Semantic Graph  $\mathcal{G} = \{V, A\}$  based on the Definition 4, where the node  $V$  of  $\mathcal{G}$  are the cell regions, and  $A$  is the adjacent matrix of  $\mathcal{G}$ .

### 4.3 Contextual network

As mentioned above, the OD matrix is very sparse as many entries are zeros. To overcome the problem of data sparsity, we propose a contextual network to effectively leverage the crowd flow features and external features which are proven to be helpful to OD prediction [23]. In the phase of flow feature extraction, we stack ConvLSTM layers to encode the spatio-temporal dependencies with the help of batch normalization [10] and Relu. Meanwhile, in the phase of external feature extraction, first we transform each external attribute into a low-dimensional vector by feeding them into different embedding layers, and then Stacked Fully Connected(FC) layers are used to model external feature. Finally, the crowd flow features and external features are integrated and form the contextual features.

**ConvLSTM** ConvLSTM combines CNN and LSTM, and is widely used in various spatio-temporal prediction tasks, such as traffic accident prediction, crowd flow prediction, and precipitation prediction. The input and hidden state of ConvLSTM in a time-stamp are all 3D tensors, and the convolution operation is conducted for both input-to-state and state-to-state connection. More specifically, ConvLSTM does the convolution operation on the data

in each time-stamp(i.e.,  $\mathcal{X}^t$ ) first, and then passes them along the time span  $[t-k+1, \dots, t]$  through LSTM module, which can be formulated as:

$$\begin{aligned} i_t &= \sigma(W_{\mathcal{X}i} * \mathcal{X}^t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i), \\ f_t &= \sigma(W_{\mathcal{X}f} * \mathcal{X}^t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f), \\ C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{\mathcal{X}c} * \mathcal{X}^t + W_{hc} * H_{t-1} + b_c), \\ o_t &= \sigma(W_{\mathcal{X}o} * \mathcal{X}^t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o), \\ H_t &= o_t \circ \tanh(C_t), \end{aligned} \quad (2)$$

where ‘\*’ denotes the convolution operator, ‘ $\circ$ ’ denotes the Hadamard product,  $\sigma$  is the logistic sigmoid function,  $i_t$ ,  $f_t$ ,  $C_t$ ,  $o_t$ , and  $H_t$  are input gate, forget gate, memory cell, output gate and hidden state, and  $W_{\alpha\beta}$  ( $\alpha \in \{\mathcal{X}, h, c\}$ ,  $\beta \in \{i, f, o, c\}$ ) are the parameters of convolutional layers in ConvLSTM.

**Fully Connected Layer** Fully Connected (FC) Layer is adopted to encode external information (e.g., Weather, Holiday and etc.) representation. The formula of FC can be represented as:

$$e_t = FC(E_t) = W_{e_t} \times E_t + b_{E_t} \quad (3)$$

where  $W_{e_t}$ ,  $b_{e_t}$  are parameter of FC layer and bias separately. Finally, the concatenation of  $H_t$  and  $e_t$  gives the final embedding for auxiliary tasks, i.e.,  $H_{con} = [H_t, e_t]$ . We denote the Contextual Network as *ContextNet*( $\cdot$ ).

#### 4.4 OD inference network

It is non-trivial to model spatial and temporal dependencies of OD data because of their variability. We propose a Seq2Seq based OD Inference Network(ODIN) is shown at the bottom of Fig. 2. First, we feed OD Image  $\mathcal{M}$  and Semantic Graph  $\mathcal{G}$  into OD Encoder to get high-dimensional OD representation  $H_{OD}$ . Second,  $H_{OD}$  and  $H_{con}$  are connected in an effective way which is denoted by  $H_{all}$  in order to tackle the problem of data sparsity. Finally,  $H_{all}$  is input into OD Decoder to predict next time-slot OD demands. In addition, the novel Hierarchical Convolutional Long and Short Term Memory(HC-LSTM) network is proposed to encode spatio-temporal embedding effectively.

##### 4.4.1 OD encoder

The OD images, and semantic graphs are input into the OD Encoder for OD feature learning. As the structures of images and graphs are totally different, they are not able to be processed by a unified neural network structure. We propose a hierarchical convolutional LSTM(HC-LSTM) network to address this problem. HC-LSTM first learns the data representations of images and graphs separately, and then fuses them together.

**HC-LSTM** As illustrated in the upper right of Fig. 2, HC-LSTM adopts stacked CNN layers and stacked GCN layers combined with LSTM model to learn the latent representations of OD images and Semantic graph. Here we use 2-dimensional convolutions on the tensors of OD images to capture the geographical spatial correlations. To more broadly capture the spatial correlations(i.e., Semantic spatial correlations), we construct semantic graph based on the OD demands among the regions, and perform graph convolutional operation. Then,



the two types of data representations are integrated which is input into a LSTM layer to learn temporal dependency. The formula of  $i$ -th HC-LSTM is shown as follows:

$$\begin{aligned} H_{conv}^i &= CNN_i(\mathcal{M}^t, W_{\mathcal{M}}), \\ H_{gcn}^i &= GCN_i(\mathcal{G}^t, W_{\mathcal{G}}), \\ H_{ST}^i &= LSTM_i(H_{conv}^i \oplus H_{gcn}^i) \end{aligned} \quad (4)$$

where  $H_{conv}^i$  and  $H_{gcn}^i$  are data representations of OD image and semantic graph respectively learned by  $i$ -th layer  $CNN_i$  and  $GCN_i$ . The  $GCN_i$  operator is as follows:

$$GCN_i(\mathcal{G}^t, W_{\mathcal{G}}) = E^t X_{\mathcal{G}}^t W_{\mathcal{G}} \quad (5)$$

where  $E^t$  is the adjacency matrix of graph  $\mathcal{G}^t$ ,  $X_{\mathcal{G}}^t$  is the graph feature.  $\oplus$  denotes the concatenation operation over  $H_{conv}^i$  and  $H_{gcn}^i$ . We create the inverse operation of the node creation over  $H_{gcn}^i$ , so that it can be concatenated with  $H_{conv}^i$ . After concatenating two types of data representations, the final representations learned through  $LSTM$  are denote as  $H_{ST}^i$ . The OD Encoder is denoted as  $ODEncoder(\cdot)$ .

#### 4.4.2 OD decoder

The learned contextual features and OD features are then input into OD Decoder to decode the data representation for prediction. As shown in the bottom right of the Fig. 2, the OD Decoder first integrates the OD features and contextual features, and then inputs the features into stacked HC-LSTM module, followed by Batch-normalization(BN) layer and Relu. First, we integrate the OD features and contextual features as follows:

$$H_{dec} = (Tanh(H_{con}))^T \times (\mathcal{M}^t + H_{ST}) \quad (6)$$

where '+' represents sum operation across channels which is also named residual operation,  $H_{ST}$  is OD feature and  $H_{con}$  is contextual feature. Then the feature  $H_{dec}$  is input into stacked HC-LSTM layer coupled with BN and Relu to learn high-dimensional representations for prediction. We denote the OD Decoder as  $ODDecoder(\cdot)$ .

#### 4.5 Overall objective function

In the final prediction step, the objective function of this task is as follows:

$$Loss = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \|\hat{Y}_i - Y_i\|^2 \quad (7)$$

where  $\mathcal{N}$  is the training sample size,  $\hat{Y}_i$  is the prediction and  $Y_i$  is the ground truth. We aim to minimize this prediction error via back-propagation and gradient descent. The pseudo-code of the algorithm is shown in Algorithm 1.

**Algorithm 1** Auxiliary-tasks enhanced spatio-temporal network.**Require:**  $\mathcal{M}$ : OD Image;  $\mathcal{X}$ : Flow Image;  $E$ : External information**Ensure:** Parameter set  $\Theta$ 

```

1: Initialize parameters  $\Theta$ 
2: while not convergence do
3:    $0 \leftarrow t$ ,  $t$  is the  $t$ -th time slot
4:   while  $t < T$  do
5:     Sample  $\{\mathcal{M}^t, Y^t\} \in \mathcal{M}, \mathcal{X}^t \in \mathcal{X}, E^t \in E$ 
6:      $H_{con}^t \leftarrow$  Crowd Flow and External information feature learning with contextual
       network  $ContextNet(\mathcal{X}^t, E^t)$ 
7:      $H_{ST}^t \leftarrow$  OD feature learning with OD Encoder  $ODEncoder(\mathcal{M}^t)$ 
8:      $\hat{Y}^t \leftarrow$  Predict OD demands with OD Decoder  $ODDecoder(H_{con}^t, H_{ST}^t)$ 
9:      $t \leftarrow t + 1$ 
10:    Update  $\Theta$  based on Objective function
11:   end while
12: end while

```

## 5 Experiment

### 5.1 Dataset and experiment setup

#### 5.1.1 Datasets

We select two large datasets which are widely used in spatio-temporal prediction for evaluation: *BikeNYC* and *TaxiNYC*. The details of the two datasets are introduced as follows.

***BikeNYC*** This dataset contains more than 9 million bike trips in New York from January 2015 to December 2015. In total, NYCBike has established over 600 bike stations and 10,000 bikes in New York. Each bike trip contains the trip duration, start/end station IDs, start/end timestamps, station Latitude/Longitude and bike ID. For this dataset, we use the first 11 months data for training, and the last month data for testing.

***TaxiNYC*** This dataset contains over 160 million taxicab trip records in New York from January 2015 to December 2015. On average, there are about 13 million trip records each month. Each taxi trip record includes fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, and etc. For this dataset, we also use the first 11 months data for training and validating, and the last month data for testing.

We also use some external features including *weather conditions, holidays and POI. Precipitation, snow, temperature and etc.*, are included in weather conditions. Whether the day is *weekday, weekend or holiday* is also considered as the people mobility patterns on holidays and regular days are quite different. The data description on the two datasets are shown in Table 1.

#### 5.1.2 Baselines

We compare the proposed AEST with the following 6 baseline methods including ARIMA, ConvLSTM [19], STResNet [29], GCRN [18], GEML [23] and MDL [31].

**Table 1** Dataset description

Dataset	<i>BikeNYC</i>	<i>TaxiNYC</i>
Data Type	Bike Trip	Taxi Trip
Latitude	40.67~40.77	40.67~40.77
Longitude	-74.02~73.95	-74.02~73.95
Time span	1/1/2015~31/12/2015	1/1/2015~31/12/2015
Time interval	1 hour	1 hour
Grid map size	(16, 16)	(16, 16)
Trajectory data		
# of trips	9 million	160 million
# of time intervals	8,754	8,754
External features		
Weather conditions	6 types (precipitation, snow, temperature, etc.)	
Days	weekday, weekend, holiday etc.	

- **ARIMA** Auto-Regressive Integrated Moving Average (ARIMA) is a classic statistic-based method for time series prediction.
- **ConvLSTM** ConvLSTM is a variant of LSTM which contains a convolution operation inside the LSTM cell. ConvLSTM considers both geographical spatial and temporal dependency of spatio-temporal data, and is widely used in many spatio-temporal prediction tasks.
- **STResNet** STResNet stacks convolutional layers and residual units to capture the spatial dependency and short/long-term temporal dependency. External features are also incorporated into STResNet.
- **GEML** Grid-Embedding based Multi-Task Learning is a multi-task learning framework that predicts the crowd flow and flow OD simultaneously similar to our work. It uses grid embedding and multi-task LSTM to capture the spatio-temporal representations.
- **MDL** MDL is a recent state-of-the-art multi-task learning framework for predicting both the node flows and edge flows on a spatial-temporal network.

To further evaluate the effectiveness of basic component of our model, we also compare the full version AEST with the following variants:

- **No-ContextNet** This model removes the contextual network. By comparing with it, we test whether the proposed *ContextNet* is useful to solve the problem of data sparsity and improve the prediction performance.
- **No-GCN** This model does not consider the features of the semantic graph. Through comparing with this model, we test whether integrating the semantic graph is helpful to capture complex spatial features.

### 5.1.3 Implement details

We implement our model with Pytorch framework on NVIDIA Tesla M40 GPU. The model parameters are set as follows. The data size of OD images is  $5 \times 16 \times 16 \times 256$ , where 5 is

the previous time slot length used for prediction,  $16 \times 16$  is the size of the cell regions, and 256 is the number of channels which represents OD demands between each region pairs.

The input flow data size is  $5 \times 16 \times 16 \times 2$ , where 2 is the numbers of channels that represents inflow and outflow. The learning rate and batch size are set to 0.0001 and 32, respectively. The structure of model output is  $1 \times 16 \times 16 \times 256$ . The baseline methods are implemented based on the original papers or we use the publicly code. Also, the set of parameters are followed by original paper.

#### 5.1.4 Evaluation metrics

Mean Absolute Error(MAE), and Root Mean Square Error(RMSE) are adopted as the evaluation metrics defined as follows:

$$MAE = \frac{1}{n} \sum_{t=1}^n \|\hat{Y}^t - Y^t\|, RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n \|\hat{Y}^t - Y^t\|^2} \quad (8)$$

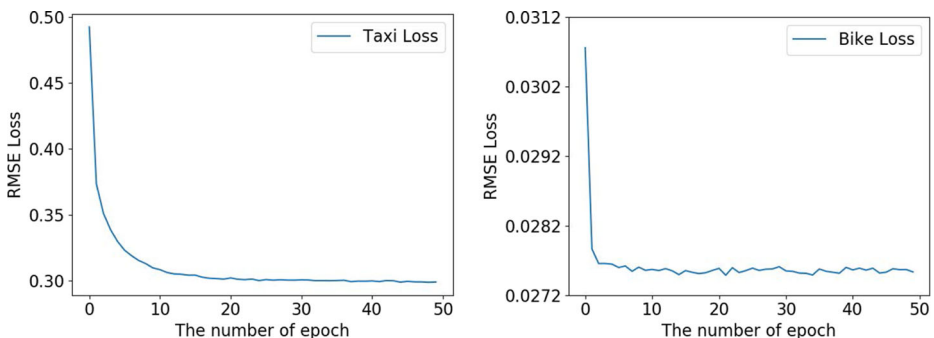
where  $\hat{Y}^t$  is the prediction, and  $Y^t$  is the ground truth.

#### 5.1.5 Loss curve

Figure 4 shows the training loss curves of the algorithm on the two datasets. one can see that the the AEST converges after about 50 epochs on both datasets which means it converges quickly. The loss curve of NYCTaxi drops smoothly, while the loss curve of NYCBike does not drop smoothly. This is mainly because the bike data is more sparse than taxi data. In the following experiment, we train AEST on both datasets with 50 epochs.

### 5.2 Comparison with baselines

Table 2 explicitly shows the performance comparison among different baselines on the two datasets. It shows that the proposed AEST achieves the best performance among all the method on both tasks which is highlighted with bold font. It shows that traditional statistics-based method ARIMA achieves the worse performance among all the methods in both cases. It is not surprising because ARIMA only uses the time series data of each region, but ignore the spatial dependency. On NYCBike dataset, compared with the best results achieved by baselines, AEST reduces RMSE of OD prediction from 0.115 (ConvLSTM) to 0.104, and



**Fig. 4** Loss curve of AEST on NYCTaxi and NYCBike

**Table 2** RMSE and MAE comparison among different methods

Model	RMSE		MAE	
	NYCBike	NYCTaxi	NYCBike	NYCTaxi
ARIMA	0.964	1.884	0.148	2.545
ConvLSTM	0.112	0.459	0.035	0.132
STResNet	0.138	0.834	0.027	0.418
GEML	0.147	0.670	0.024	0.136
MDL	0.154	1.103	0.041	0.950
AEST	<b>0.104</b>	<b>0.454</b>	<b>0.021</b>	<b>0.126</b>

MAE from 0.024(GEML) to 0.021, respectively. On NYCTaxi dataset, AEST improves the RMSE from 0.459 (ConvLSTM) to 0.456, and MAE from 0.132 (ConvLSTM) to 0.126. The RMSE and MAE of NYCBike are much smaller than NYCTaxi, because that the bike trips are much sparser than taxi trips. In addition, the OD of taxi trips can be anywhere in the city, while the OD of bike trips is fixed(e.g., bike stations). The results in Table 2 show that the proposed AEST is superior to existing state-of-the-art spatio-temporal learning approaches.

### 5.3 Comparison with variation models

To study the effect of different components in AEST on the model performance, we conduct experiments by comparing AEST with its variants No-ContextNet, and No-GCN. The result is shown in Table 3. One can see that the ContextNet and GCN are useful to the model in that losing any one of them will increase the prediction error. On both datasets, ContextNet seem more important which supports our point of view that incorporating flow information will be helpful to solve the problem of data sparsity and improve model performance. In addition, the semantic graph is also useful for both datasets. Combining these components together achieves the lowest RMSE and MAE, demonstrating that all of them are useful to the studied problem.

### 5.4 Case study on prediction vs ground truth

To further intuitively illustrate how accurately our model can predict OD demands, we visualize the predicted OD demands and ground truth in two figures as depicted in Figs. 5 and 6. Due to the data sparsity of NYCBike data, we show the case study on NYCTaxi. The Fig. 5 shows the OD demands from  $r_{8,14}$  to  $r_{6,12}$ . One can see that the prediction curve can accurately trace the ground truth curve which demonstrates the effectiveness of the proposed

**Table 3** RMSE and MAE comparison with variant methods

Dataset	Methods	RMSE	MAE
NYCBike	No-ContextNet	0.108	0.032
	No-GCN	0.107	0.023
	AEST	0.104	0.021
NYCTaxi	No-ContextNet	0.465	0.130
	No-GCN	0.456	0.127
	AEST	0.454	0.126

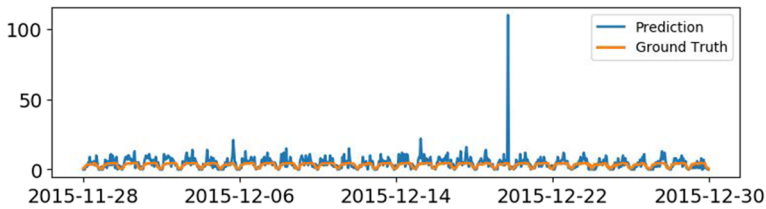


Fig. 5 OD demands prediction vs ground truth:  $r_{8,14} \rightarrow r_{6,12}$

model. The temporal trend of OD demands is also well understood by our model. Our model can perfectly capture the periodicity of the data, which is largely due to the usage of auxiliary tasks. However, it is obvious from the picture 5 that our model is not good at capture the sudden changes of OD demands. To further demonstrate the superiority of the proposed model, we show the case study on OD Matrix prediction vs ground truth at different time-slots on December 30, 2015. The OD matrix with the size  $C \times C$  (e.g.,  $256 \times 256$ ) is converted by OD image, where  $C = m \times n$  is the number of all regions. We choose four time-slots which are 8:00 am, 10:00 am, 14:00 pm, and 18:00 pm respectively. From the picture 6, the prediction of OD matrix very matches the ground truth. The results shows the proposed AEST model can effectively predict OD demands across the city.

### 6 Conclusion and future work

In this paper, we proposed a novel Auxiliary-tasks Enhanced Spatio-Temporal Network(AEST) to predict OD demands via learning crowd flow and external information as auxiliary tasks. The novelty of the model lies in the usage of contextual network to facilitate OD prediction. An end-to-end solution is proposed to effectively learning sufficient auxiliary features for OD prediction to address the data sparsity issue. To effectively capture the complex spatial temporal dependency, a Hierarchical Convolutional LSTM(HC-LSTM) is

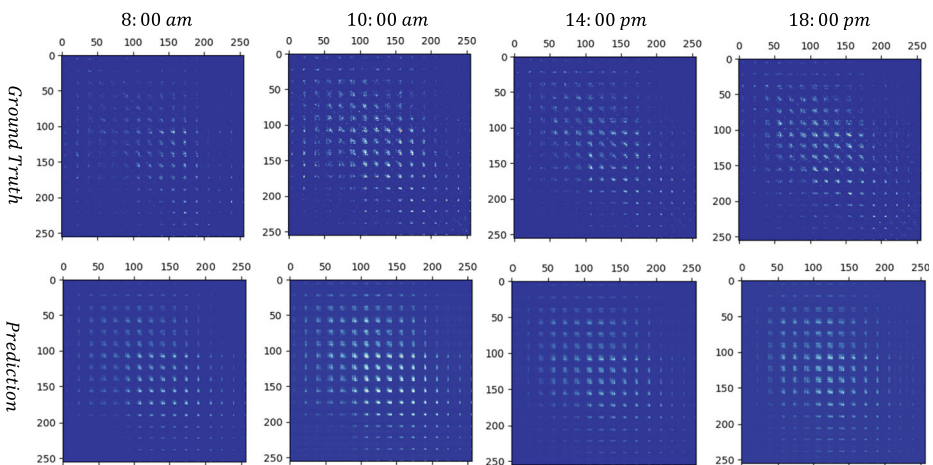


Fig. 6 OD Matrix prediction vs ground truth at different time-slots on December 30, 2015(left to right: 8:00 am, 10:00 am, 14:00 pm, 18:00 pm)

designed. We evaluate the proposed model on two real large datasets collected from New York. The results demonstrate the superior performance of the model on OD prediction.

In the future, we will explore how to design a more accurate model to capture sudden changes of OD demands. It also would be interesting to further apply the proposed model to more spatio-temporal tasks in different application scenarios such as traffic prediction, crime prediction and traffic accident detection. There are three reasons for the model generalization. First, the two data formats of OD and the other spatial temporal data(e.g., traffic data) are similar. We can convert the OD data into spatial maps or graphs. Second, the contextual information is helpful to spatial temporal prediction, such as traffic prediction, which is demonstrated in previous works [28]. Third, the proposed model containing HC-LSTM and ConvLSTM is general for all spatial-temporal tasks to learn the spatial and temporal representations.

**Acknowledgements** This work is supported by National Key R&D Program of China (No.: 2018YFB1003900), CCF-Tencent Open Research Fund and the Fundamental Research Funds for the Central Universities (No.: NZ2020014).

## References

1. Ashok K, Ben-Akiva ME (2002) Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Trans Sci* 36(2):184–198
2. Bierlaire M, Crittin F (2004) An efficient algorithm for real-time estimation and prediction of dynamic od tables. *Oper Res* 52(1):116–127
3. Cetin M, Comert G (2006) Short-term traffic flow prediction with regime switching models. *Transp Res Rec* 1965(1):23–31
4. Chen Y, Xiao D (2009) Traffic network flow forecasting based on switching model. *Control and Decision* 24(8):1177–1180
5. Chu KF, Lam AY, Li VO (2019) Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions. *IEEE Transactions on Intelligent Transportation Systems*
6. Diao Z, Wang X, Zhang D, Liu Y, Xie K, He S (2019) Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 33, pp 890–897
7. Du B, Peng H, Wang S, Bhuiyan MZA, Wang L, Gong Q, Liu L, Li J (2020) Deep irregular convolutional residual lstm for urban traffic passenger flows prediction. *IEEE Trans Intell Transp Syst* 21(3):972–985. <https://doi.org/10.1109/TITS.2019.2900481>
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
9. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
10. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. [arXiv:150203167](https://arxiv.org/abs/1502.03167)
11. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. [arXiv:160902907](https://arxiv.org/abs/1609.02907)
12. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
13. Lee S, Fambro D (1999) Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board* 1678
14. Li Y, Yu R, Shahabi C, Liu Y (2017) Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. [arXiv:170701926](https://arxiv.org/abs/1707.01926)
15. Lin Z, Feng J, Lu Z, Li Y, Jin D (2019) Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 33, pp 1020–1027
16. Liu L, Qiu Z, Li G, Wang Q, Ouyang W, Lin L (2019) Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Trans Intell Transp Syst* 20(10):3875–3887

17. Ren J, Xie Q (2017) Efficient od trip matrix prediction based on tensor decomposition. In: 2017 18Th IEEE international conference on mobile data management, MDM, IEEE, pp 180–185
18. Seo Y, Defferrard M, Vandergheynst P, Bresson X (2018) Structured sequence modeling with graph convolutional recurrent networks. In: International conference on neural information processing, Springer, pp 362–373
19. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc (2015) Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
20. Wang S, Cao J, Chen H, Peng H, Huang Z (2020) Seqst-gan: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6(4):1–24
21. Wang S, Cao J, Yu P (2020) Deep learning for spatio-temporal data mining: a survey. *IEEE Trans Knowl Data Eng*, pp 1–20. <https://doi.org/10.1109/TKDE.2020.3025580>
22. Wang S, Miao H, Chen H, Huang Z (2020) Multi-task adversarial spatial-temporal networks for crowd flow prediction. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 1555–1564
23. Wang Y, Yin H, Chen H, Wo T, Xu J, Zheng K (2019) Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1227–1235
24. Williams B (2001) Multivariate vehicular traffic flow prediction: Evaluation of arimax modeling. *Transportation Research Record: Journal of the Transportation Research Board* 1776
25. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Zhenhui L (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: The Thirty-Second AAAI conference on artificial intelligence
26. Yao H, Tang X, Wei H, Zheng G, Li Z (2019) Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 5668–5675
27. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. [arXiv:170904875](https://arxiv.org/abs/1709.04875)
28. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the Thirty-First AAAI conference on artificial intelligence, AAAI Press, pp 1655–1661
29. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. *Proceedings of AAAI*
30. Zhang J, Chen F, Wang Z, Liu H (2019) Short-term origin-destination forecasting in urban rail transit based on attraction degree. *IEEE Access* 7:133452–133462
31. Zhang J, Zheng Y, Sun J, Qi D (2019) Flow prediction in spatio-temporal networks based on multitask deep learning. *IEEE Transactions on Knowledge and Data Engineering*
32. Zhang Y, Wang S, Chen B, Cao J, Huang Z (2019) Trafficgan: Network-scale deep traffic prediction with generative adversarial nets. *IEEE Trans Intell Transp Syst*, pp 1–12. <https://doi.org/10.1109/TITS.2019.2955794>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



## Affiliations

Hao Miao<sup>1</sup> · Yan Fei<sup>2</sup> · Senzhang Wang<sup>1,3</sup> · Fang Wang<sup>4</sup> · Danyan Wen<sup>5</sup>

Hao Miao  
miaohao@nuaa.edu.cn

Yan Fei  
201961120003@njtech.edu.cn

Fang Wang  
fangwang@bipt.edu.cn

Danyan Wen  
wendy2018@njust.edu.cn

- <sup>1</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China
- <sup>2</sup> College of Computer Science and Technology, Nanjing Tech University, Nanjing, 211816, China
- <sup>3</sup> Collaboration Innovation Center of Novel Software Technology and Industrialization, Nanjing, 211106, China
- <sup>4</sup> School of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing, 102617, China
- <sup>5</sup> School of Economics and Management, Nanjing University of Science and Technology, Nanjing, 210094, China