Urban Flow Pattern Mining Based on Multi-Source Heterogeneous Data Fusion and Knowledge Graph Embedding

Jia Liu, Tianrui Li, *Senior Member, IEEE*, Shenggong Ji, Peng Xie, Shengdong Du, Fei Teng, and Junbo Zhang, *Member, IEEE*

Abstract—Urban flow analysis is an essential research for smart city construction, in which urban flow pattern analysis focuses on the continuous state of urban flow. How to mine, store and reuse traffic patterns from urban multi-source heterogeneous big data is challenging. Therefore, this paper proposes a knowledge mining network for regional flow pattern to mine and store the urban flow pattern. The proposed model consists of two modules. In the first module, the features of the region and its flow pattern are extracted as the entity and relation, respectively. In the second module, POI features are modeled to enhance the embedding representation of relation and entity. Based on the translation distance method, the knowledge triplets of regional flow patterns are mined. Finally, the proposed model is compared with some benchmark methods using Chengdu Didi order and POI datasets. Experimental results show that the proposed model is effective. In addition, the knowledge triplets are visualized and some application examples are introduced.

Index Terms—Data mining, urban flow pattern, knowledge graph embedding, multi-source heterogeneous data fusion, urban computing

1 Introduction

The innovation of big data-based smart city technology has a great impact on the development and operation of smart city. The urban computing, which is a process of collecting, fusing and analyzing urban multi-source heterogeneous big data [1], is effective and valuable for solving problems in the city. For example, some models based on machine learning have been proposed to predict urban traffic flow [2], [3], [4], crowd flow prediction [5], [6], urban air quality [7], water quality [8], and weather [9], and so on. These methods can efficaciously predict the outcome of certain future moments in the city. More intelligently, some methods can achieve real-time prediction [10], [11].

In particular, urban flow analysis is an important application in urban computing, which has gained a lot of attention from researchers. However, the results obtained by existing flow analysis methods are instantaneous, so only the discrete state of the city can be analyzed, as shown in Fig. 1a. Flow pattern, such as morning peak and evening peak, is a representation of the continuous flow state of a city. For urban flow analysis, an important study is to analyze the flow pattern between specific regions of the city. Specifically, urban flow pattern analysis is the study of the flow change trend between specific regions of the city, as shown in Fig. 1b.

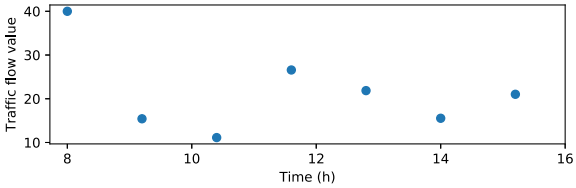
Generally, in a city there are many same traffic flow states at all times, such as morning peak and evening peak during diverse weekdays. Zhang *et al.* used similar characteristic of urban traffic flow, i.e. trend, period and closeness to optimize the flow prediction results [12]. If the state of the city can be preserved, we can directly obtain the flow information of the city based on the features of the same state. Fortunately, combined with the knowledge storage and high-speed feedback capability of the knowledge graph, which is a set of fact triples consisting of head entity h, relation r and tail entity t, i.e., $KG = \{(h, r, t)\}$, it is of great significance to achieve the mining, storage and reuse of the urban flow pattern.

Therefore, the goal of this paper is to learn urban traffic features from urban multi-source heterogeneous data and treat them as entities and relations respectively, and then construct knowledge triplets of traffic patterns with the help of knowledge graph embedding method, so as to realize the mining of the urban flow patterns.

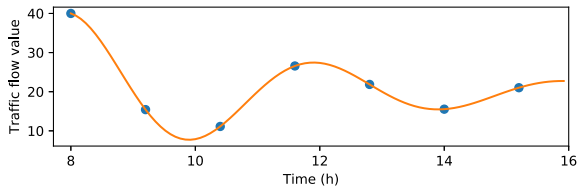
The analysis of urban flow patterns has the following significances. First, it can effectively avoid unnecessary prediction of regional flow in the similar state. For example, when we are required to know whether there is an early peak traffic situation between region A and region B from 8:00 to 9:00 today, we only need the knowledge of the traffic flow trend between regions A and B at a certain time period

- *Jia Liu, Tianrui Li, Peng Xie, Shengdong Du, and Fei Teng are with the Institute of Artificial Intelligence, School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, Sichuan 611756, China. E-mail: xiaoke92@foxmail.com, {trli, sddu, fteng}@swjtu.edu.cn, pengxie@my.swjtu.edu.cn.*
- *Shenggong Ji is with the Tencent Inc., Shenzhen, Guangdong 518054, China. E-mail: shenggongji@163.com.*
- *Junbo Zhang is with Urban Computing Business Unit, JD Finance, Shenzhen, Guangdong 518054, China. E-mail: msjunbozhang@outlook.com.*

Manuscript received 7 Dec. 2020; revised 12 July 2021; accepted 15 July 2021. Date of publication 21 July 2021; date of current version 10 Jan. 2023.
This work was supported by the National Key R&D Program of China under Grant 2019YFB2101802 and the National Natural Science Foundation of China under Grant 61773324. Data source: Didi Chuxing.
(Corresponding author: Tianrui Li.)
Recommended for acceptance by M. A. Cheema.
Digital Object Identifier no. 10.1109/TKDE.2021.3098612



(a) Discrete state of flow in the city.



(b) Continuous state of flow in the city.

Fig. 1. Discrete and continuous states of traffic flow in city.

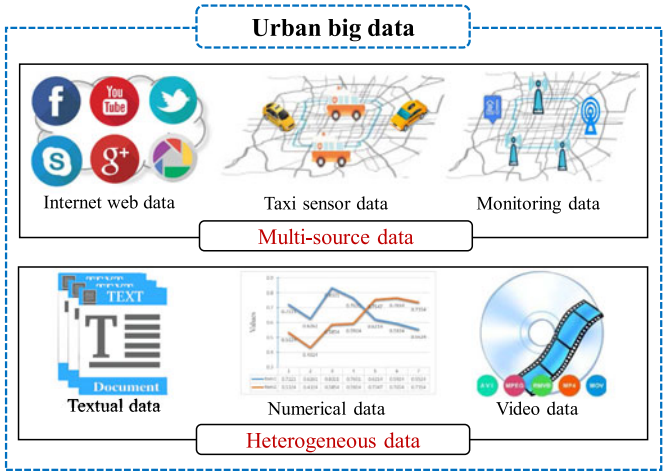


Fig. 2. Urban multi-source heterogeneous data. [14]

(including information from 8:00 to 9:00) to make a decision on the early peak situation, instead of using a traffic prediction model to make an accurate prediction of the current traffic flow value. Second, it can be employed in commercial siting and urban planning applications by classifying the embedding of regions, as shown in Section 5.6. Third, the urban flow pattern can be applied as an auxiliary feature of the urban air quality prediction to improve the accuracy of the prediction. Fourth, it can also be employed as a criterion for determining abnormal flow changes in urban regions.

Urban flow pattern analysis mines features from multi-source heterogeneous data and stores them as knowledge triples, which faces the following two challenges.

(1) Urban Multi-Source Heterogeneous Data Analysis. Urban multi-source heterogeneous data is shown in Fig. 2. Multi-source data includes Internet web data, taxi sensor data, monitoring data, etc., and heterogeneous data includes textual data, numerical data, video data, and so on. Urban multi-source heterogeneous data analysis is an indispensable but arduous task for the construction of smart cities. First, it is difficult to fully consider data from different sources, types and meanings. Second, the output of deep learning cannot be effectively explained, which is an obstacle to the fusion of multi-source heterogeneous data. In this case, how temporal feature of traffic flow and spatial feature of region can be effectively extracted, fused and stored from these data is a technical challenge worth studying. Third, most of the existing urban flow forecasting works consider that urban flow is influenced by many factors. For example, weather and other factors are considered in the citywide crowd flow prediction, and these factors are used as auxiliary information [13]. Likewise, regional traffic pattern requires considering various factors.

(2) Extraction of Knowledge Triples From Numeric Data. The common method of knowledge mining is to extract fact triples from textual data. The resulting entities and relations generally exist in the form of text, such as the fact triplet $<Da\ Vinci, painted, Mona\ Lisa>$. However, urban big data that has a lot of hidden information exists in the form of numbers, so existing knowledge mining methods are not suitable for extracting knowledge triples from numerical data. Moreover, the elements of traditional knowledge triples are stored in the form of words. How should the elements of knowledge triples extracted based on numerical data be stored0 Because it cannot be stored in a way similar to text-based storage. In this paper, we mainly analyze and extract the features of numerical data, and then mine its knowledge. The mined knowledge is stored in the form of vector, which is the result of knowledge graph embedding relative to the traditional knowledge mining method.

In this paper, we mine regional flow pattern in city as knowledge via the fusion of multi-source heterogeneous data and knowledge graph embedding, and propose a knowledge mining model for regional flow pattern. In summary, the main contributions are summarized as follows:

- A knowledge mining network for regional flow pattern (RFP-KMN) is proposed to construct the knowledge triplet of urban flow pattern. The proposed method respectively extracts temporal feature and spatial feature of multi-source heterogeneous data as relation and entity through different Autoencoders, and fuses and stores them via knowledge graph embedding techniques.
- Compared with the traditional knowledge triple mining from textual data, this paper proposes to store the mined features of multi-source heterogeneous data as knowledge triples.
- Generally, the translation distance-based method is used to embed knowledge triples. In this paper, this method is used to construct the knowledge triples of region flow pattern from the embedding representation of entities and relations. To the best of our knowledge, this is the first attempt to use the classical knowledge embedding representation to construct the knowledge triples from the embedding representation of entities and relations.
- The proposed model and some benchmark methods are compared through Chengdu Didi order data and point-of-interest (POI) data, and the knowledge of regional flow pattern is visualized.

The remainder of this paper is organized as follows. In the next section, we present the related work about knowledge graph embedding and urban knowledge graph. In Section 3, we give an overview of the proposed RFP-KMN model. Section 4 details the knowledge mining of regional flow pattern.

The experiment result, visualization and analysis are given in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

2.1 Knowledge Graph Embedding

Knowledge graph embedding is a method of embedding entities and relations into a continuous vector space. It encodes the interaction between entities and relations through a specific model architecture to achieve knowledge representation learning [15], [16], [17]. Knowledge graph embedding methods are roughly divided into translating-based methods [18], [19], [20], matrix factorization-based methods [21], [22], deep learning-based methods [23], [24], [25] and graph neural network-based methods [26].

The translating-based methods evaluate the rationality of fact triples by the distance between two entities. For triples (h, r, t), these methods get $h + r \approx t$ as much as possible by analyzing the scoring function. The TransE model embedded the entities and relationships in the same space, and then regarded the embedding representation of the relation r as a translation between the embedding representations of the head entity h and the tail entity t [18]. The TransH model mapped the head entity and tail entity to the hyperplane corresponding to the relation r, so that the entities had different embedding representations under different relations [19]. It can handle one-to-many, many-to-many, and many-to-many relations. The TransG model expressed multiple semantics of a relationship through Gaussian distribution, and proposed a knowledge graph embedding method based on Gaussian mixture model [20]. This model can solve the problem that entities cannot be distinguished due to fuzzy relation semantics.

The matrix factorization-based methods usually express the properties of the knowledge graph in a matrix format, and factorize the matrix to obtain the embedding representation of nodes and relations. For a matrix V representing the properties of nodes, the structure information of V can be embedded through the factorization of V, i.e., $V = V_e M V_e^T$, where $V_e \in \mathbb{R}^{|v| \times k}$ is an embedding matrix, which is the embedding representation of the node. $M \in \mathbb{R}^{k \times k}$ is an interaction matrix, and k specifies the number of potential features. By solving its objective function, the optimal embedding representation of the node can be derived. Matrix factorization is used in knowledge graph embedding. From the view of time, space and spatio-temporal, Zhuang *et al.* constructed a knowledge graph with nodes and edges as addresses and their relations [21]. They extracted the embedding representations of nodes in time and space through matrix factorization. Padia *et al.* used matrix factorization to encode prior knowledge and empirically explored different methods, and proposed a knowledge graph fact prediction method based on knowledge-rich matrix factorization [22]. These methods first construct a matrix of specific knowledge graph properties, and then factorize to obtain the corresponding embedding representation. In addition, matrix factorization can be further improved by imposing non-negativity constraints on the factors to obtain a better embedding representation.

Deep learning has great advantages in learning new feature representations, so it is widely used for embedding representations of knowledge graph. The embedding performance of the knowledge graph will be constrained by the shallow fast model, because increasing the size of the embedding is the only way to increase the number of features in the shallow model, which is difficult to scale to a large-scale knowledge graph. A model ConvE using 2D convolution on embedding was proposed to predict the link, which was defined by a single convolution layer, a projection layer of embedding dimension and an inner product layer [23]. In view of the disadvantage that ConvE only considered the local relationship among different dimensional entries, the ConvKB model took triplets as a matrix of three columns as input, and learned the representation of each column through Conv1D to obtain an embedding representation of triplet elements [24]. Based on ConvKB, the CapsE model reconstructed these knowledge graphs into corresponding capsules, and then routed the capsule to another capsule to generate a continuous vector [25]. The length of this vector is used to measure the triple credibility. In addition, graph neural network is the most popular research in graph-based deep learning, and it is widely applied in the research of knowledge graph embedding with standard graph structure. Xie *et al.* established a model based on a heterogeneous graph neural network, which contains two types of nodes: sentence nodes and entity nodes [26]. They employed convolutional neural networks to capture the context information of the nodes.

2.2 Urban knowledge graph

Smart city technology is the collection, integration, sharing and analysis of city multi-source heterogeneous big data combined with machine learning. It is used to study the characteristics of the city and solve the existing problems. The urban knowledge graph is an important task in smart city technology, which can realize the mining, storage and analysis of urban knowledge. The existing urban knowledge graph construction is mainly to extract entity and relation from urban data and form knowledge triples. In order to detect urban emergency events in real time, Xu *et al.* proposed a knowledge base-based urban emergency event detection model, which is a crowdsourcing-based urban knowledge base model [27]. Based on a large amount of urban multi-source spatio-temporal data, Zhao *et al.* presented a multi-source spatio-temporal data analysis framework for knowledge graph embedding and gave a definition of urban knowledge graph [28]. They used the directed network $CKG = (K, G)$ to describe the urban knowledge graph, where G is the travel network and K is the knowledge network that describes the auxiliary information set. In addition, some existing POI recommendation work are also studied through the urban knowledge graph. Zhuang *et al.* proposed a framework for constructing urban movement knowledge graph with spatial and temporal information [21]. This framework can effectively predict users' attention to different POIs in the city. Yang *et al.* developed a social and sequence-aware next POI recommendation model [29]. The model used an improved heterogeneous network to learn better POI representation that contains social relationships and geographic information.

Urban multi-source heterogeneous data needs to be well represented and fused. This is the key to mine urban

knowledge. On the one hand, the existing urban knowledge graph construction method primarily extracts fact triples from urban textual data, such as translation-based POI recommendation [30] and graph embedding for city problem analysis [31], [32]. These methods do not analyze the potential features of urban multi-source heterogeneous data, and cannot extract knowledge from numerical data. On the other hand, other knowledge mining methods lack the effective fusion of urban multi-source heterogeneous data to gain knowledge more accurately, such as the framework for constructing urban movement knowledge graph [21].

3 Overview

In this section, we discuss our proposed method, including some preliminaries, problem formulation and an introduction of the proposed RFP-KMN model.

3.1 Preliminaries

Definition 1. *(Region set): According to different granularities and semantics, a region has many different definitions [12]. In this study, we divide the effective area of a city into an $M \times N$ grid map according to latitude and longitude. A grid with POI (Point of Interest) attributes represents a region v_i, where POI attributes can represent the semantics of region ($\mathbf{P} \in \mathbb{R}^{|v|\times|v|}$ is the regional POI semantic similarity matrix, shown in Section 4.3. $|v|$ indicates the number of the regions). All regions constitute the semantic region set V.*

Definition 2. *(Transfer): A traffic transfer Tr_{ij} represents a transfer from the trajectory point p_s (starting point) to the trajectory point p_d (destination), where each trajectory point p consists of {time stamp, latitude and longitude}. Since the trajectory points p_s and p_d belong to regions v_i and v_j, respectively. Tr_{ij} can be expressed as $Tr_{ij} = \{v_i \rightarrow v_j\}$. By calculating the transfer of all regions in different time, a time series transfer matrix $\mathbf{Tr} \in \mathbb{R}^{|v|^2\times T}$ is obtained. $\mathbf{Tr}_{it}$ means the flow of the i^{th} region pair in the period of time $t-1$ to t. It is used for R modeling. Moreover, by counting the regional transfer of the entire time period, the regional transfer tensor $\mathbf{U} \in \mathbb{R}^{|v|\times M\times N}$ [1] is formed. It is used for V modeling.*

Definition 3. *(Relative distance): Relative distance Rd_{ij} is the Euclidean distance between the regions v_i and v_j according to the divided $M \times N$ grid map. We can construct a region relative distance tensor $\mathbf{D} \in \mathbb{R}^{|v|\times M\times N}$ through counting the distance among all areas. It is used for V modeling.*

Definition 4. *(Regional flow pattern set): The regional flow pattern r represents the trend of regional flow between two regions over time. The R is the set of regional flow pattern, where $r_i \in R$ represents a regional flow pattern. The features of flow trends among regions are learned and used as the regional flow pattern r. That is, it models the relations in the knowledge graph (R modeling).*

Definition 5. *(Knowledge graph of regional flow pattern): The knowledge graph is represented as $G(V, R)$, where V is a region set with POI semantics, and R is a set of regional flow patterns $r(v_i, v_j)$ between regions v_i and v_j. After learning, each region v and regional flow pattern r are represented in a vector format.*

1. $|v| = M * N$ is hold. To clearly show the shape of the data, such as the 3D region transfer tensor $\mathbf{U} \in \mathbb{R}^{|v|\times M\times N}$, we do not directly use $M * N$ to denote the number of regions $|v|$.

3.2 Problem Formulation

Given the time series transfer matrix **Tr**, the regional transfer tensor **U** and the relative distance tensor **D**, regional POI semantic similarity matrix **P**, learn the knowledge triple $< v_i, v_j, r >$ of regional flow pattern, where $v_i, v_j \in V$, $r \in R$. Firstly, the embedding representation of R is learned through **Tr**, and the embedding representation of V is learned through **U** and **D**. Secondly, **P** is used to learn for enhancing the embedding representation of V and R. Finally, the knowledge triple $< v_i, v_j, r >$ is learned by modeling R and V using translation distance method.

3.3 The Proposed Model

In order to mine knowledge triple of regional flow pattern from urban multi-source heterogeneous big data, we propose a knowledge mining network for regional flow pattern (RFP-KMN) to mine regional flow pattern. Our method mainly includes two modules. In the first module, we embed the relation set and entity set in the knowledge graph through the deep learning models, namely, relation extraction and entity extraction. It includes two steps, which are the urban big data preprocessing, and relation and entity extraction. In the second module, the knowledge embedding representation method is used to model the relation set and entity set to mine the knowledge triple of regional flow pattern. It includes two steps, which are data fusion and knowledge triple mining. Fig. 3 shows the architecture of RFP-KMN.

In the first module, for the urban big data preprocessing, we preprocess the regional transfer data, map data, and regional POI data into a time series transfer matrix **Tr** and a regional transfer tensor **U**, a relative distance tensor **D**, and a regional POI semantic similarity matrix **P**, respectively. The detailed preprocessing methods are illustrated in Section 4. For the relation extraction and entity extraction, we use four autoencoders to model the **Tr**, **U**, **D** and **P** respectively. Specifically, LSTM Autoencoder is adopted to model regional flow patterns (R modeling) and 2D-CNN Autoencoder is designed to model regional features (V modeling). 1D-CNN Autoencoder is applied to model regional POI semantic to enhance representation learning of R and V through data fusion.

In the second module, for the data fusion, regional POI features are used to enhance the embedding representation of relations and entities. For relation enhancement, each relation r is a regional flow pattern between specific two regions, so the weights of POI feature among diverse regions should be different when fusing regional POI features. Therefore, an attention-based fusion method is proposed to fuse relation and regional POI features. For entity enhancement, we first obtain entity V by fusing the spatial transfer features and spatial location features of the region, and then use concatenate fusion to add POI features. For the knowledge triple mining, we use knowledge embedding representation method to mine knowledge triple from R and V. Because the translation distance method (TransE [18]) is simple and effective, it is used to model R and V.

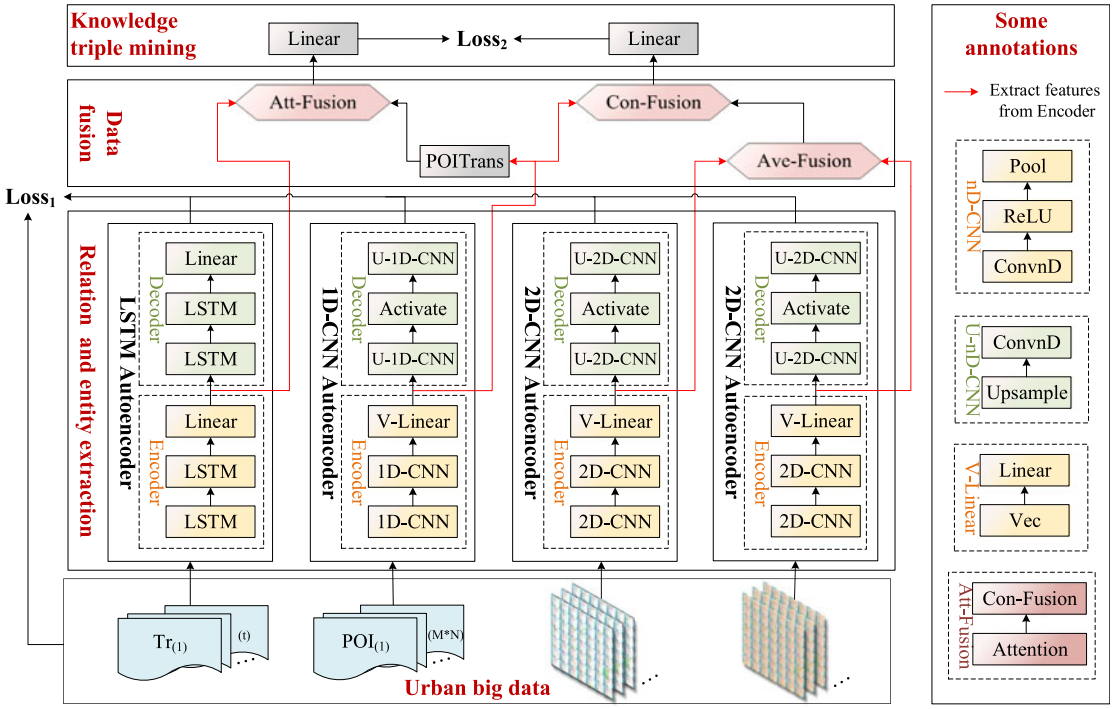


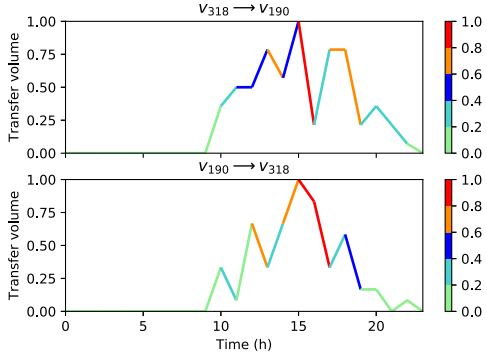
Fig. 3. The architecture of RFP-KMN consists of two modules. The first module is to extract relation and entity (Loss_1), including urban big data preprocessing, relation extraction (R modeling) and entity extraction (V modeling). The second module is to mine the knowledge triple of regional flow pattern (Loss_2), including data fusion and knowledge triple mining. Specifically, regional flow patterns are modeled by LSTM Autoencoder (R modeling), and region features are modeled by 2D-CNN Autoencoder (V modeling). Regional POI semantic is modeled by 1D-CNN Autoencoder, which is used to enhance representation learning of R and V through data fusion. And the knowledge of regional flow pattern is constructed through a translation distance modeling method (TransE [18]). In particular, the Vec layer is used to change the dimension of the output of the previous layer.

4 Knowledge Mining of Regional Flow Pattern

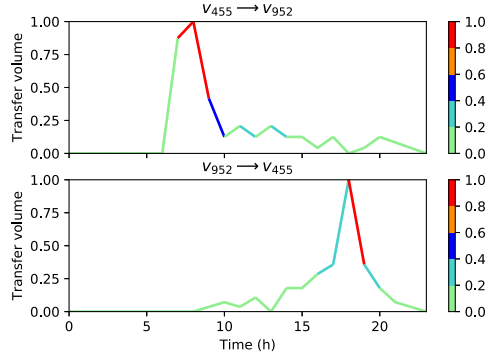
4.1 Regional Flow Pattern Modeling (R Modeling)

Traffic transfer Tr_{ij} can be easily obtained by mapping each trajectory p to the corresponding region v_i. By adding time feature, a time series transfer matrix $\mathbf{Tr} \in \mathbb{R}^{|v|^2 \times T}$ is constructed, where $|v|$ indicates the number of the regions, $|v|^2$ represents the number of region transfer, and T represents the time. The transfer matrix $\mathbf{Tr}$ has a region-to-self transfer, namely Tr_{ii}.

Figs. 4a and 4b show the traffic transfer in regions v_{318} and v_{190} and the traffic transfer in regions v_{455} and v_{952}, respectively. The POI features and relative distances of the four regions are shown in Table 1. From Table 1, we can get some features of these regions: Regions v_{318} and v_{190} have subway station, office building and residence, and have close relative distance; Regions v_{455} and v_{952} have subway station and close relative distance, but they are residence and office building respectively. As shown in Fig. 4a, the transfer patterns of regions v_{318} and v_{190} are similar from a global perspective. However, from a local perspective, the difference in transfer volume among time periods 10-11, 12-13, and 15-16 is large, and the difference among time periods 9-10, 14-15, and 18-19 is small. Since regions v_{318} and v_{190} have subway stations, office buildings, and residences, the traffic transfer between them changes greatly. Compared with the regions v_{318} and v_{190}, the transfer patterns of the regions v_{455} and v_{952} are different, as shown in Fig. 4b. The relative POI features of regions v_{455} and v_{952} are residence and office building respectively, so the traffic transfer volume $Tr_{455,952}$ appears very large in the time period 6-9 (the morning rush hour), and the traffic transfer volume $Tr_{952,455}$ appears very large in the time



(a) Transfer of regions v_{318} and v_{190}.



(b) Transfer of regions v_{455} and v_{952}.

Fig. 4. The region-region traffic transfer volume within a day. The transfer volume is normalized. Each colored line represents the transfer volume from time t to $t+1$. Among them, the red line indicates that there is a lot of traffic transfer from time t to $t+1$.

TABLE 1
POI Details and Relative Distance of the Region

Region	Subway	Bus	Residence	Office	Park	Relative distance
v_{318}	✓	✓	✓	✓	✓	3.1623 ($Rd_{318\leftrightarrow 190}$)
v_{190}	✓		✓	✓		3.1623 ($Rd_{318\leftrightarrow 190}$)
v_{455}	✓	✓	✓		✓	5.0000 ($Rd_{455\leftrightarrow 952}$)
v_{952}	✓			✓	✓	5.0000 ($Rd_{455\leftrightarrow 952}$)

The POI can represent the semantics of region, and the relative distance can show the relevance of two regions on the map. $Rd_{i\leftrightarrow j}$ is the relative distance between the regions v_i and v_j.

period 17-19 (the evening rush hour). Through observation, flow patterns between regions are unique and trendy.

LSTM is a special kind of RNN that can process sequence data effectively [33]. Because of its superiority in processing sequence information, LSTM is widely used in text analysis, speech analysis, and big data analysis with temporal features, such as natural language processing [34], speech recognition [35], and spatiotemporal prediction models [36], [37]. Regional flow pattern is the change trend of traffic volume based on time characteristic. Therefore, LSTM Autoencoder is used to extract the features of traffic flow. The LSTM Autoencoder module consists of two parts: encoder and decoder. The encoder projects the time series transfer matrix $\mathbf{Tr} \in \mathbb{R}^{|v|^2 \times T}$ into the hidden unit space, and the decoder projects the hidden unit to the time series transfer matrix.

The encoder consists of two LSTM structures, followed by a linear layer. In our work, the decoder and encoder have the same structure. Inputting the time series transfer matrix into the encoder, the transfer features of any two regions will be obtained in a specific time period, which is expressed as $R \in \mathbb{R}^{|v|^2 \times k}$, where k represents the feature dimension of the time series transfer matrix. Each transfer feature $r(v_i, v_j)$ $(r \in R)$ represents the traffic flow change feature of regions v_i and v_j in a specific time period. It is called the flow pattern of regions v_i and v_j. In addition, the POI features of the region are added to enhance the embedding representation of regional flow patterns. Its detail can be found in Section 4.4. Compared with the triples in the traditional knowledge graph, the knowledge $< v_i, v_j, r >$ does not have its inverse relation, that is $< v_i, v_j, r^- >$.

4.2 Region Embedding Modeling (V Modeling)

Region v_i (head entity) and region v_j (tail entity) in knowledge $< v_i, v_j, r >$ are correlated. There are two kinds of relations between them, namely reversible relation and irreversible relation. In this paper, the relative distance between regions is used to represent their reversible relation, while the transfer volume between regions is used to indicate their irreversible relation. For example, the transfer volume from region v_{318} to region v_{190} is 81, and the transfer volume from region v_{190} to region v_{318} is 88. And the relative distance between them is the same, that is $Rd_{318\leftrightarrow 190} = 3.1632$. The embedding representation of the region is achieved by effectively embedding and fusing the relations between the regions. In addition, POI information is added to increase the region semantics, which can enhance the embedding representation of the region. Its detail can be found in Section 4.4. Region transfer U_{ij} can be easily obtained through the statistics of the traffic transfer Tr_{ij}, and finally a region transfer tensor $\mathbf{U} \in \mathbb{R}^{|v| \times M \times N}$ is formed, as shown in Fig. 5.

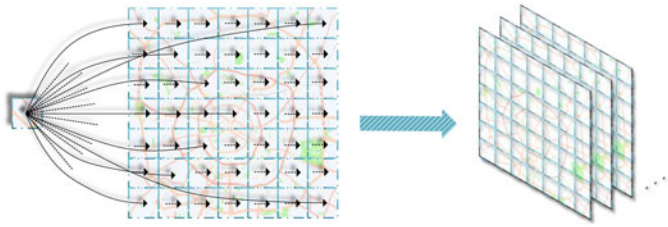


Fig. 5. The region transfer tensor $\mathbf{U} \in \mathbb{R}^{|v| \times M \times N}$, which shows the regional irreversible relation.

CNN is a deep learning method that extracts data features through convolution kernel. Because CNN has strong expressive ability, it is widely used in video analysis, image analysis and big data analysis with spatial features, such as person re-identification [38], human ear recognition [39], urban spatio-temporal prediction model [3], [13]. In this paper, we use 2D-CNN Autoencoder to extract the features of region transfer. Similar to the LSTM Autoencoder module, the 2D-CNN Autoencoder module consists of two elements: encoder and decoder. The encoder projects the region transfer tensor $\mathbf{U} \in \mathbb{R}^{|v| \times M \times N}$ into the hidden unit space, and the decoder projects the hidden unit to the region transfer tensor.

The encoder consists of two 2D-CNN layers and one V-Linear layer. Among them, 2D-CNN layer is composed of Conv2D layer, ReLU layer and Pool layer, and V-Linear layer is composed of Vec layer and Linear layer. The Vec layer is used to change the dimension of the output of the previous layer. The decoder consists of two U-2D-CNN layers and one activation layer. Among them, U-2D-CNN layer consists of an Upsample layer and a Conv2D layer. The network structure of 2D-CNN Autoencoder is shown in Fig. 3. The region transfer tensor $\mathbf{U} \in \mathbb{R}^{|v| \times M \times N}$ is inputted into the encoder, and the region's transfer features can be obtained and expressed as $V_{tr} \in \mathbb{R}^{|v| \times k_2}$, where k_2 represents the feature dimension of the region transfer tensor.

The relative distance Rd_{ij} between the regions can be obtained by calculating the divided grid map. Finally, we can construct a region relative distance tensor $\mathbf{D} \in \mathbb{R}^{|v| \times M \times N}$, as shown in Fig. 6. Since the relative distance is the spatial features of the region, we also use 2D-CNN Autoencoder to extract the features of the region space. The network structure of 2D-CNN Autoencoder used to train $\mathbf{D}$ is the same as that used to train $\mathbf{U}$. By inputting the region relative distance tensor $\mathbf{D} \in \mathbb{R}^{|v| \times M \times N}$ to the encoder, the region relative distance feature $V_{rd} \in \mathbb{R}^{|v| \times k_3}$ is obtained, where k_3 represents the feature dimension of the region relative distance tensor. In order to easily fuse two types of relation features among regions, we set $k_2 = k_3$ in this paper.

4.3 Regional POI Semantic Modeling

Regional traffic transfer is commonly affected by the regional POI features, so the regional POI features are extracted to enhance the semantic embedding representation of the region. In addition, the regional flow pattern is also influenced by regional POI features, thus it is also used to reinforce the embedding representation of the regional flow pattern. Encoding the POIs of the region, we obtain the regional initial POI feature matrix $\mathbf{P}^{ini} \in \mathbb{R}^{|v| \times k_4}$, where P_i^{ini}

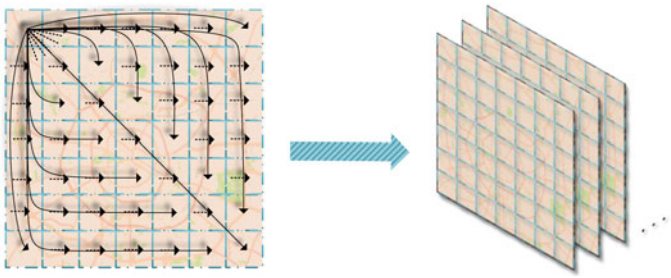


Fig. 6. The region relative distance tensor $\mathbf{D} \in \mathbb{R}^{|v| \times M \times N}$, which illustrates the regional reversible relation.

represents the initial POI feature vector of the region v_i and k_4 represents the initial POI feature dimension. The POI semantic similarity matrix P_{ij} of a region will be obtained by calculating the POI similarity of the regions v_i and v_j, and finally a regional POI semantic similarity matrix $\mathbf{P} \in \mathbb{R}^{|v| \times |v|}$ is formed. Considering that POI features are one-dimensional features and do not have temporal features, 1D-CNN Autoencoder is considered to extract regional POI features. Compared to 2D-CNN Autoencoder, Conv1d is used for the convolutional layer in 1D-CNN Autoencoder. By encoding the POI semantic similarity matrix $P \in \mathbb{R}^{|v| \times |v|}$ through the encoder, the POI semantic feature matrix $\mathcal{P} \in \mathbb{R}^{|v| \times k_2}$ will be obtained, where k_2 represents the feature dimension of the POI semantic matrix of the region. The $p_i (p_i \in \mathcal{P})$ represents the POI semantic feature of region v_i.

4.4 Data Fusion

For the fusion of regional flow patterns and regional POI semantic features, attention mechanism is used to specifically strengthen the POI semantic features of specific two regions. It is called attention-based fusion. The $\mathcal{P} \in \mathbb{R}^{|v| \times k_2}$ obtained from the 1D-CNN Autoencoder model is the POI features of regions. In order to fuse it with the regional flow pattern between the two regions, we first convert it to the change of POI features between regions through the POITrans layer. The attention-based fusion is mainly divided into two layers, i.e. the Attention layer and the Concatenate fusion layer. The regional POI semantic features are input to the Attention layer behind the POITrans layer. Given a query $\mathcal{Q}$ and a set of key-value $\mathcal{K}$ and $\mathcal{V}$, the attention scores are computed by using a dot-product to compute unnormalized saliencies.

$$Attention = softmax\left(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d}}\right)\mathcal{V}, \tag{1}$$

where d is the dimension of the key vectors served as a scaling factor. The regional POI semantic features with attention are fused with the regional flow pattern through the concatenate fusion layer. Regional flow pattern $R \in \mathbb{R}^{|v|^2 \times k}$ will be obtained after a Linear layer.

For fusing region features and regional POI semantic features, the relative distance features V_{rd} and traffic transfer features V_{tr} among regions are fused by average fusion layer. Next, the regional POI semantic features are fused through concatenate fusion layer. The embedding representation of regions $V \in \mathbb{R}^{|v| \times k}$ will be obtained after a Linear layer.

4.5 Loss function

In the first module, our goal is to obtain embedding representations of relation and entity through deep learning models. For LSTM Autoencoder, 1D-CNN Autoencoder and two 2D-CNN Autoencoder models, we minimize the mean square error of the input samples (x_i) and output values (y_i):

$$Loss_1 = \frac{1}{\mathfrak{N}} \sum_i \|x_i - y_i\|_2^F, \tag{2}$$

where $Loss_1$ is used as the loss function of the LSTM Autoencoder and nD-CNN Autoencoder (n=1 or 2) models, and $\mathfrak{N}$ is the number of samples. $\|\cdot\|_2^F$ indicates the Frobenius norm.

In the second module, our target is to mine the knowledge triple of regional flow pattern using the translation distance modeling method. The knowledge triplet $< v_i, v_j, r >$ can be extracted by the regional embedding representation V and the regional flow pattern R, where v_i, v_j belong to V and r belongs to R. The TransE model treats the relation in the knowledge graph as translation vector between entities [18]. For each fact triple $< h, r, t >$, the TransE model first embeds the entity and relation, and then regards the relation r as the translation between the head entity h and the tail entity t, that is $h + r \approx t$. The idea of the TransE model is used to construct triple knowledge. Given a triple set S as a positive sample set, we randomly generate a negative sample set S'.

$$S' = \{(h', r, t) \ | h' \in V\} \cup \{(h, r, t') \ | t' \in V\}, \tag{3}$$

where h' and t' represent the head entity and tail entity in the generated negative sample triplet S', respectively. Each negative sample is obtained from a triple in the positive sample set S by randomly replacing the head entity or the tail entity with other entities. We define a mapping matrix Q to map the regions into the space of regional flow pattern.

$$\begin{aligned} Loss_2 = & \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} \\ & [\|hQ + r - tQ\|_2^F - \|h'Q + r - t'Q\|_2^F + \gamma]_+ \\ & + \lambda \|Q\|_2^F, \end{aligned} \tag{4}$$

where $Q \in \mathbb{R}^{k \times k}$ is the mapping of the region vector on the regional flow pattern space. $\gamma > 0$ is a hyper parameter served as a margin between positive triplet and negative triplet, and $\lambda \|Q\|_2^F$ is a regularization term to prevent overffitting. $[x]_+$ is the positive part of the x, i.e. $[x]_+ = max\{0, x\}$. By minimizing the objective function $Loss_2$, Q is learned.

5 Experiments

In this section, to validate the effectiveness of our proposed RFP-KMN model, we compare its performance to some baselines on the real-world traffic flow dataset in Chengdu. All experiments are conducted in Python 3.5 with torch 0.4.1 on a PC Server, and the server configuration is Intel(R) Core(TM) i7-8700 CPU 3.20GHz, 4 GPUs each is 12G NVIDIA Tesla K80C, and memory is 128GB.

5.1 Datasets

The datasets we use include Didi order data in Chengdu, Chengdu POI data and POI lookup table. Chengdu order

data is provided by Didi Chuxing GAIA Initiative.[2] Since the GCJ-02 coordinate system is used by the latitude and longitude in the trajectory data, the POI lookup table used is provided by Gaode Development Platform.[3] The trajectory data is processed by routing, which ensures that the data can correspond to the actual road information. The total number of order data is approximately 250,000*30. Each piece of data includes order id, time and latitude and longitude of getting on and off.

5.2 Hyperparameters

The hyperparameter settings of the first module are shown below. For the 1D-CNN Autoencoder network, the convolutions of the encoder use 64 filters of size 3×3 and 128 filters of size 4×4. In the decoder, the size scaling factors are 8 and 3 respectively, and the convolutions uses 64 filters of size 3×3 and 1 filters of size 2×2. For the 2D-CNN Autoencoder network, the convolutions of the encoder uses 64 filters of size 3×3 and 128 filters of size 3×3. In the decoder, the size scaling factors are 8 and 2 respectively, and the convolutions use 64 filters of size 3×3 and 1 filters of size 6×6. The Adam [40] is used for optimization in four antoencoder networks, and the batch size is 125. In addition, the value range of the learning rate in LSTM Autoencoder is [0.001, 0.004], and the value range of the learning rate in nD-CNN Autoencoder is [0.1, 0.4]. In particular, the training process of the four networks is carried out simultaneously during the experiment.

The hyperparameter settings of the second module are shown below. The learnable parameter Q is initialized using the uniform distribution of the default parameters in Pytorch. The values of γ and λ in the TransE-based model are 1.0 and 0.25, respectively. The SGD (Stochastic Gradient Descent) is used in the TransE-based model, and the batch size is 100. The value range of the learning rate in the TransE-based model is [0.01, 0.04].

5.3 Evaluation mechanism

Similarity of Entities/Relations. In order to evaluate the utility of the knowledge triplet of regional flow pattern, we compare the similarity of entities on different days and the similarity of relations on different days. The regional flow pattern represents the flow transfer trend between two regions within a certain period of time. Relative to other regions, the regional flow pattern $r(v_i, v_j)$ on days day_1 and day_2 should be similar. Therefore, we measure the utility of R in the knowledge of regional flow pattern by calculating the similarity of regional flow patterns between the same two regions on different days. On the other hand, for the entity set V, the difference in the embedding representation of the regions on different days is caused by the irreversible relations of the regions. Therefore, we equivalently calculate the similarity of the embedding representation of the same region on different days to measure the utility of V in the knowledge of regional flow pattern.

Cosine similarity evaluates the similarity of two vectors by calculating the angle cosine of the two vectors. We use it to separately calculate the similarity of entities and relations on different days. The greater the similarity of entities or relations, the better the utility of the knowledge of regional flow pattern.

$$Sim_R = \frac{1}{|R|} \sum_{r_{ik} \in R_{dayi},\, r_{jk} \in R_{dayj}} \frac{r_{ik} \cdot r_{jk}}{\|r_{ik}\|_2^F \times \|r_{jk}\|_2^F}, \tag{5}$$

$$Sim_V = \frac{1}{|V|} \sum_{v_{ik} \in V_{dayi},\, v_{jk} \in V_{dayj}} \frac{v_{ik} \cdot v_{jk}}{\|v_{ik}\|_2^F \times \|v_{jk}\|_2^F}, \tag{6}$$

where r_{ik} and r_{jk} represent the k^{th} relation on day i and the k^{th} relation on day j, respectively. v_{ik} and v_{jk} indicate the k^{th} entity on day i and the k^{th} entity on day j, respectively. R_{dayi} and V_{dayi} separately represent the relation set and entity set on the i^{th} day, and $|R|$ and $|V|$ indicate the number of elements in relation set and entity set, respectively.

Link Prediction. In addition, to evaluate the knowledge construction method of regional flow pattern, we use the same sorting procedure as in [18], [41] to perform link prediction task. For each constructed triple, the head/tail entity is removed and replaced by all entities in turn. The score of the newly formed triplet is calculated by $\|h'Q + r - t'Q\|_2^F$ and sorted in an ascending order. Using the same scoring function to calculate the link prediction results, we can evaluate the quality of the input data, namely V and R. We evaluate the knowledge construction method by calculating the average ranking of the correct entity and its proportion in top 5, 10 and 20 respectively, i.e. $hits@5$, $hits@10$ and $hits@20$. In this task, the input is all the knowledge triples and the output is all the tail entities in ascending order with head entity and relation pair as the target.

Flow Prediction: An Application Example of Urban Flow Pattern. Moreover, in order to better evaluate the regional flow patterns learned by different models, we use diverse regional flow patterns to train a logistic regression model to predict the flow at different time between diverse regions. It is worth noting that the RFP-KMN method proposed in this paper stores traffic flow patterns as knowledge triples. When predicting traffic flow, the traffic flow pattern r in the knowledge triple $< v_i, r, v_j >$ is used to predict the traffic flow between region v_i and region v_j. Existing traffic flow prediction methods use urban big data to predict, which requires more complex models and longer training time. To predict traffic flow through traffic flow patterns, only simple prediction model is needed. Therefore, in the experiment, we use the logistic regression model to predict the traffic flow when using the traffic flow pattern to predict the traffic flow. In addition, in order to compare the effect of prediction, we apply some time series forecasting models to predict traffic flow by training raw urban big data. The evaluation indicators Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are employed to evaluate the predicted results. The experimental process of flow prediction is shown in Fig. 7.

5.4 Baselines

The existing urban knowledge graph construction methods principally extract entity and relation from textual data to construct knowledge triples. These triples are not embedded, so tasks such as link prediction cannot be used to evaluate the performance of the constructed knowledge graph.

2. https://gaia.didichuxing.com
3. https://lbs.amap.com/

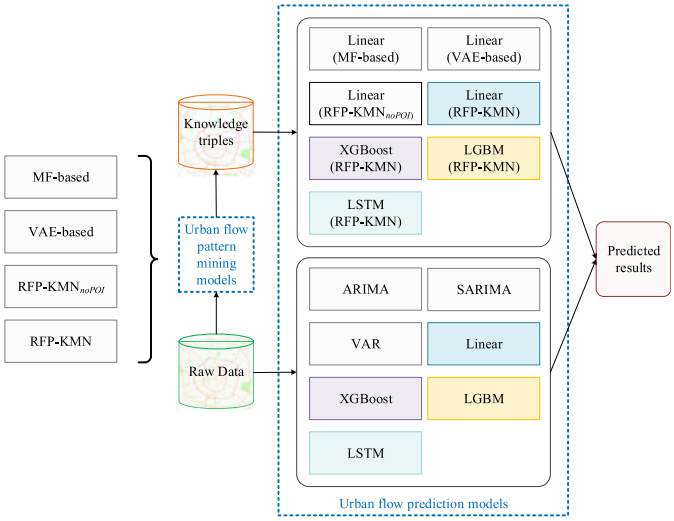


Fig. 7. The experimental process of flow prediction. **Input:** The raw data (traffic flow transfer **Tr**) or the traffic flow pattern in the knowledge triple (obtained through the traffic flow pattern mining models). It is worth noting that the inputs of LGBM and XGBoost are the raw data and its mean, variance, skewness and kurtosis. **Model:** Traffic flow prediction models, see Section 5.4 for details. The same background color in the figure refers to the same model with different inputs. **Output:** The RMSE and MAE values that evaluate the predict performance of each model.

However, the method proposed in this paper is to extract the embedding representation of entity and relation from multi-source heterogeneous data. Therefore, tasks such as link prediction can be used to evaluate the performance of the constructed knowledge graph. In addition, although a data analysis framework carried out embedding representation of knowledge triples, the knowledge embedding representation was not involved in the construction of knowledge triples [28]. This method uses knowledge graph embedding method to learn embedding representation of knowledge triples. In general, these methods cannot be compared fairly with the proposed method. Therefore, the baselines and our models are outlined as follows.

Linear Regression Model. This model uses time characteristics for linear regression.

ARIMA (Auto-Regressive Integrated Moving Average). The ARIMA model, which is one of the most commonly used methods in time series forecasting, consists of an autoregressive model and a moving average model.

SARIMA (Seasonal Auto-Regressive Integrated Moving Average). This model is an extension of ARIMA, which supports univariate time series data with seasonal components.

VAR (Vector Auto-Regressive). The VAR model can capture the paired relationship between all streams.

XGBoost [42]. It is a scalable end-to-end tree boosting system. Since XGBoost requires suitable features as input, we use historical data and its mean, variance, skewness and kurtosis as the input features of XGBoost for prediction. We use GridSearchCV to automatically adjust some of the hyperparameters.

LGBM [43]. It is a gradient boosting framework that uses tree-based learning algorithm. We set the input of LGBM to be the same as XGBoost. And we also use GridSearchCV to automatically adjust some of the hyperparameters.

LSTM. The model is a special kind of RNN that can process sequence data effectively. When predicting traffic flow, the structure of the model is the same as the decoder structure of the LSTM Autoencoder in the proposed RFP-KMN.

Linear (RFP-KMN). The model presented in this paper. The model includes R modeling, V modeling, POI features modeling, data fusion and TransE-based triple modeling. The logistic regression model is applied to predict traffic flow.

Linear (RFP-KMN$_{noPOI}$). RFP-KMN model without POI features modeling. The model includes R modeling, V modeling, data fusion (fusing the relative distance features V_{rd} and traffic transfer features V_{tr}), and TransE-based triple modeling. The logistic regression model is applied to predict traffic flow.

Linear (RFP-KMN$_{noRd}$). RFP-KMN model without region reversible feature modeling. The model includes R modeling, V modeling (irreversible feature modeling), POI features modeling, data fusion, and TransE-based triple modeling. The logistic regression model is applied to predict traffic flow.

Linear (MF-based method) [21]. This is a method based on matrix factorization. Matrix factorization is the decomposition of a matrix into the product of several matrices. R, V_{tr}, V_{rd} and $\mathcal{P}$ are extracted by matrix decomposition of Tr, U, D and P, respectively. For tensors U and D, we change the dimensions of them firstly. Given a matrix T, the information of T is embedded through the decomposition of T, i.e., $T = \mathfrak{T}M\mathfrak{T}^T$, where $\mathfrak{T} \in \mathbb{R}^{|x|\times k}$ is an embedding matrix, and $|x|$ represents the number of region transfer or the number of regions. $M \in \mathbb{R}^{k\times k}$ represents the interaction matrix, and k represents the number of potential features of matrix T. By solving the loss function, $\mathfrak{T}$ and M can be obtained.

$$Loss_{MF} = \|T - \mathfrak{T}M\mathfrak{T}^T\|_2^F + \lambda_{MF}\left[\|\mathfrak{T}\|_2^F + \|M\|_2^F\right], \tag{7}$$

where λ_{MF} is a regularization term. MF-based method includes R modeling, V modeling, POI features modeling, data fusion, and TransE-based triple modeling. The logistic regression model is applied to predict traffic flow.

Linear (VAE-Based Method) [44]. This is a variational autoencoder based method. In this method, we use four variational encoders with the same structure and different parameters instead of LSTM Autoencoder, 1D-CNN Autoencoder, and two 2D-CNN Autoencoders, respectively. VAE-based method includes R modeling, V modeling, POI features modeling, data fusion, and TransE-based triple modeling. The logistic regression model is applied to predict traffic flow.

LSTM (RFP-KMN). The model presented in this paper. The model includes R modeling, V modeling, POI features modeling, data fusion and TransE-based triple modeling. The LSTM is applied to predict traffic flow.

XGBoost (RFP-KMN). The model presented in this paper. The model includes R modeling, V modeling, POI features modeling, data fusion and TransE-based triple modeling. The XGBoost is applied to predict traffic flow. We use GridSearchCV to automatically adjust some of the hyperparameters.

LGBM (RFP-KMN). The model presented in this paper. The model includes R modeling, V modeling, POI features

TABLE 2
Similarity of Relations on Different Days

Models	$\text{days}_{1,2}$	$\text{days}_{1,3}$	$\text{days}_{1,6}$	$\text{days}_{1,7}$	$\text{days}_{1,8}$	days_{avg}
MF-based	0.6617	0.6434	0.5618	0.5648	0.6729	0.6216
VAE-based	0.7135	0.6770	0.6941	0.6906	0.7056	0.6961
RFP-KMN$_{noPOI}$	0.7689	0.7794	0.7386	0.7292	0.7645	0.7562
RFP-KMN	0.8614	0.8428	0.8204	0.8129	0.8652	**0.8405**

$\boldsymbol{days}_{i,j}$ represents day i and day j, where $\boldsymbol{days}_{1,2}$ and $\boldsymbol{days}_{1,3}$ represent working days, $\boldsymbol{days}_{1,6}$ and $\boldsymbol{days}_{1,7}$ represent weekends, $\boldsymbol{days}_{1,8}$ represent the i-day after a week, i.e. $i = j - 7$, and $\boldsymbol{days}_{avg}$ represent the average of all days.

modeling, data fusion and TransE-based triple modeling. The LGBM is applied to predict traffic flow. And we also use GridSearchCV to automatically adjust some of the hyperparameters.

5.5 Quantitative Comparison

Table 2 illustrates the similarity of the relations among different days. We analyze the experimental results from the table longitudinally and horizontally. From the longitudinal direction of the Table 2, the similarity of the MF-based method is lower than other methods. This method has certain limitations for the extraction of time-series features. More, the VAE-based method has better performance than MF-based and worse performance than the model proposed in this paper. From the similarity value of RFP-KMN$_{noPOI}$, it can be seen that the POI features of the regions have a great influence on the flow change between the regions. Intuitively, the flow change in a region is usually related to its POI features. The proposed RFP-KMN model fuses POI features, which makes the embedding representation of the relations achieve a better result. From the horizontal direction of the Table 2, weekends ($\text{days}_{1,6}$ and $\text{days}_{1,7}$) also have a certain impact on the regional flow changes. However, only the flow changes in some regions can be affected by holidays in practice. The POI features of these regions are more similar to the POI features of other regions. Therefore, the change of regional flow is affected by various factors.

We analyze the experimental results in Table 3 in the same way. From the longitudinal direction of the Table 3, the values of similarity for MF-based method, VAE-based method and RFP-KMN$_{noRd}$ are relatively low. The reversible feature (Rd) of a region is a stable feature that can enhance the embedding representation of the region. From the result of RFP-KMN, it can be seen that the POI features of the region can also enhance the embedding representation of the region. In contrast, POI features have a greater impact on relations than entities. This also implicitly reflects that the regional flow change trend is greatly affected by POI features. From the horizontal direction of Table 3, weekends ($\text{days}_{1,6}$ and $\text{days}_{1,7}$) have little effect on the embedding representation of the region.

TABLE 3
Similarity of Entities on Different Days

Models	$\text{days}_{1,2}$	$\text{days}_{1,3}$	$\text{days}_{1,6}$	$\text{days}_{1,7}$	$\text{days}_{1,8}$	days_{avg}
MF-based	0.7289	0.7765	0.7126	0.7212	0.7257	0.7330
VAE-based	0.7615	0.8324	0.7120	0.7323	0.7439	0.7582
RFP-KMN$_{noPOI}$	0.8599	0.9124	0.8025	0.8099	0.8267	0.8415
RFP-KMN$_{noRd}$	0.7745	0.8305	0.7115	0.7214	0.7812	0.7638
RFP-KMN	0.8728	0.9460	0.8711	0.8761	0.8781	**0.8888**

$\boldsymbol{days}_{i,j}$ represents day i and day j, where $\boldsymbol{days}_{1,2}$ and $\boldsymbol{days}_{1,3}$ represent working days, $\boldsymbol{days}_{1,6}$ and $\boldsymbol{days}_{1,7}$ represent weekends, $\boldsymbol{days}_{1,8}$ represent the i-day after a week, i.e. $i = j - 7$, and $\boldsymbol{days}_{avg}$ represent the average of all days.

TABLE 4
The Results of Link Prediction

Models	Mean rank	**hits@5**(%)	**hits@10**(%)	**hits@20**(%)
MF-based	943	15.8	27.1	47.7
VAE-based	883	17.2	24.9	51.3
RFP-KMN$_{noPOI}$	484	30.2	37.4	62.2
RFP-KMN$_{noRd}$	402	37.8	45.4	50.2
RFP-KMN	**312**	**44.5**	**51.9**	**64.5**

In the link prediction task, a lower average rank and a higher hit rate indicate better link prediction results. Table 4 shows the link prediction results of different models. The proposed RFP-KMN model is better than the other models in four metrics. In addition, in the results of RFP-KMN$_{noPOI}$, the value of $hits@20$ is much larger than the value of $hits@10$, which indicates that there are a part of the regions that are not greatly affected by POI features.

In the flow prediction task, the flow prediction results of different models are shown in Fig. 8. When predicting traffic flow by raw data, LGBM has the best prediction result, with XGBoost having the second best prediction result. The prediction results of LGBM and XGBoost models with raw data as input can be used as benchmarks for the prediction results of the models with traffic flow patterns as input. We present our analysis in the following three aspects.

First, LGBM (RFP-KMN) and XGBoost (RFP-KMN) have better prediction results than the LGBM that has the best result among the models using the original data as input. Furthermore, Linear (RFP-KMN) had slightly better prediction result than XGBoost that has the second best result among the models using raw data as input. This demonstrates that RFP-KMN is able to effectively mine and reuse the traffic flow pattern from the raw data.

Then, compared to the models LSTM, Linear, LGBM and XGBoost, which use the original data as input, the corresponding LSTM (RFP-KMN), Linear (RFP-KMN), LGBM (RFP-KMN) and XGBoost (RFP-KMN) all have better prediction results. Particularly, Linear (RFP-KMN) and XGBoost (RFP-KMN) have significantly improved prediction performance. Therefore, the traffic flow patterns mined by using RFP-KMN are effective.

Finally, for the diverse traffic flow pattern mining models, the prediction result of Linear(RFP-KMN) is better than the result of Linear(RFP-KMN$_{noPOI}$), indicating that the flow in different time periods between regions is affected by the regional POI features, and the regional POI features can improve the flow prediction results. The prediction result of Linear(MF-based) and Linear(VAE-based) is not as good as the Linear(RFP-KMN) model, showing that Linear(RFP-KMN) is more suitable for extracting the features of time series data. More, the prediction result of LSTM (RFP-KMN) is not as good as that of Linear (RFP-KMN). The possible reason for this is that the traffic flow patterns are already trained by the deep learning-based model, which can lead to overfitting if the LSTM is further used.

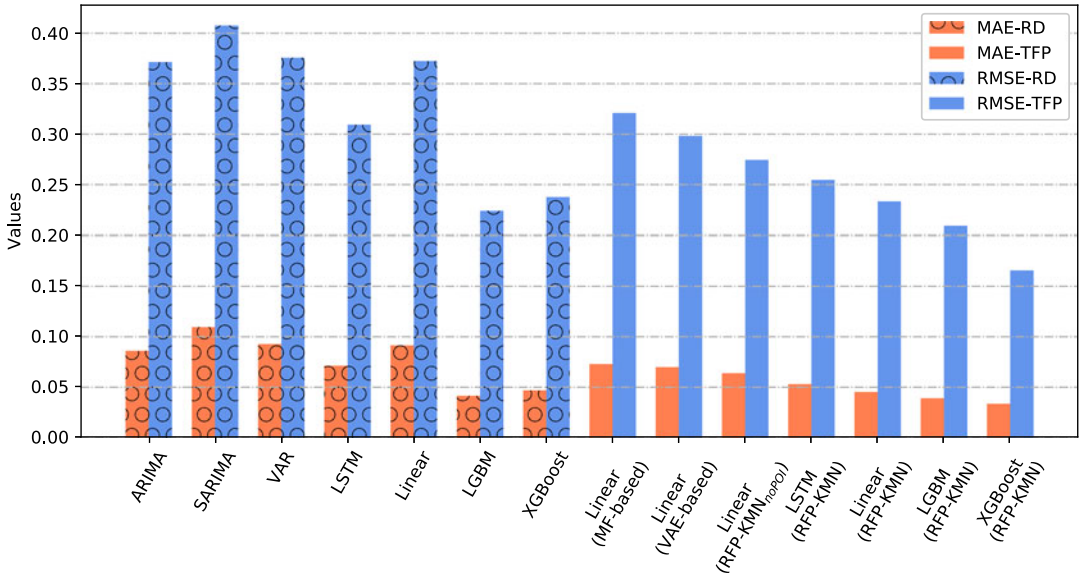


Fig. 8. The results of flow prediction. "-RD" indicates that the input data of the model is raw data (RD), and "-TFP" shows that the input data of the model is traffic flow pattern (TFP).

5.6 Visualization

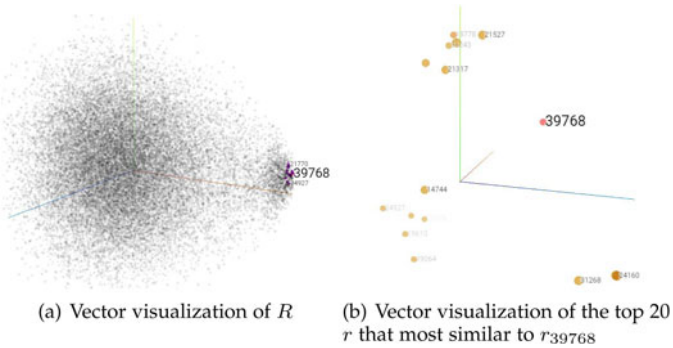
The Visualization of R Vector. We visualize the R vector, as shown in Fig. 9. Firstly, 40,000 relation vectors are selected for visualization. Secondly, the principal component analysis (PCA) [45] algorithm is used to reduce the dimensionality of the relation vector. Finally, the relation vector after dimensionality reduction is visualized. The visualization of 40,000 relation vectors is shown in Fig. 9a. In order to better display the relation vector, we randomly select r_{39768} and the top 20 relation vectors most similar to it for visualization, as shown in Fig. 9b.

Clustering Application of R. There are some rules in the relations of the general knowledge graph, such as reversible relation. The relations in the knowledge graph constructed in this paper are relatively independent, but there are certain connections among them. If the two relations have a high similarity relative to the other relations, that is, the flow patterns in these regions are similar. Fig. 10 illustrates the top five relations that are most similar to the relation r_{sample}, denoted by ①, ②, ③, ④, and ⑤ respectively. The POI features of the head and tail entities in the knowledge where these relations belong to are similar to the head and tail entities of the knowledge where r_{sample} belongs to, respectively. Putting these knowledge with similar relations into one category, we can control the flow patterns of the whole regions. Assuming that r_{sample} has the characteristic of severe congestion in the morning peak, we can formulate mitigation strategies for congestion problem similar to r_{sample} in the entire regions in advance. Therefore, we use clustering methods (e.g. k-means clustering, density-based spatial clustering) to cluster R, so as to achieve the clustering of knowledge triplet. An evaluation criterion is defined to evaluate the result of clustering.

$$Score_{cluster} = \frac{1}{|C|} \sum_{r_i, r_j \in C} (Sim_C), \tag{8}$$

where $Score_{cluster}$ is the evaluation score of the clustering result, C is all categories, and $|C|$ is the number of categories. Sim_C is a method of calculating the similarity of relations in categories in the same way as Sim_R.

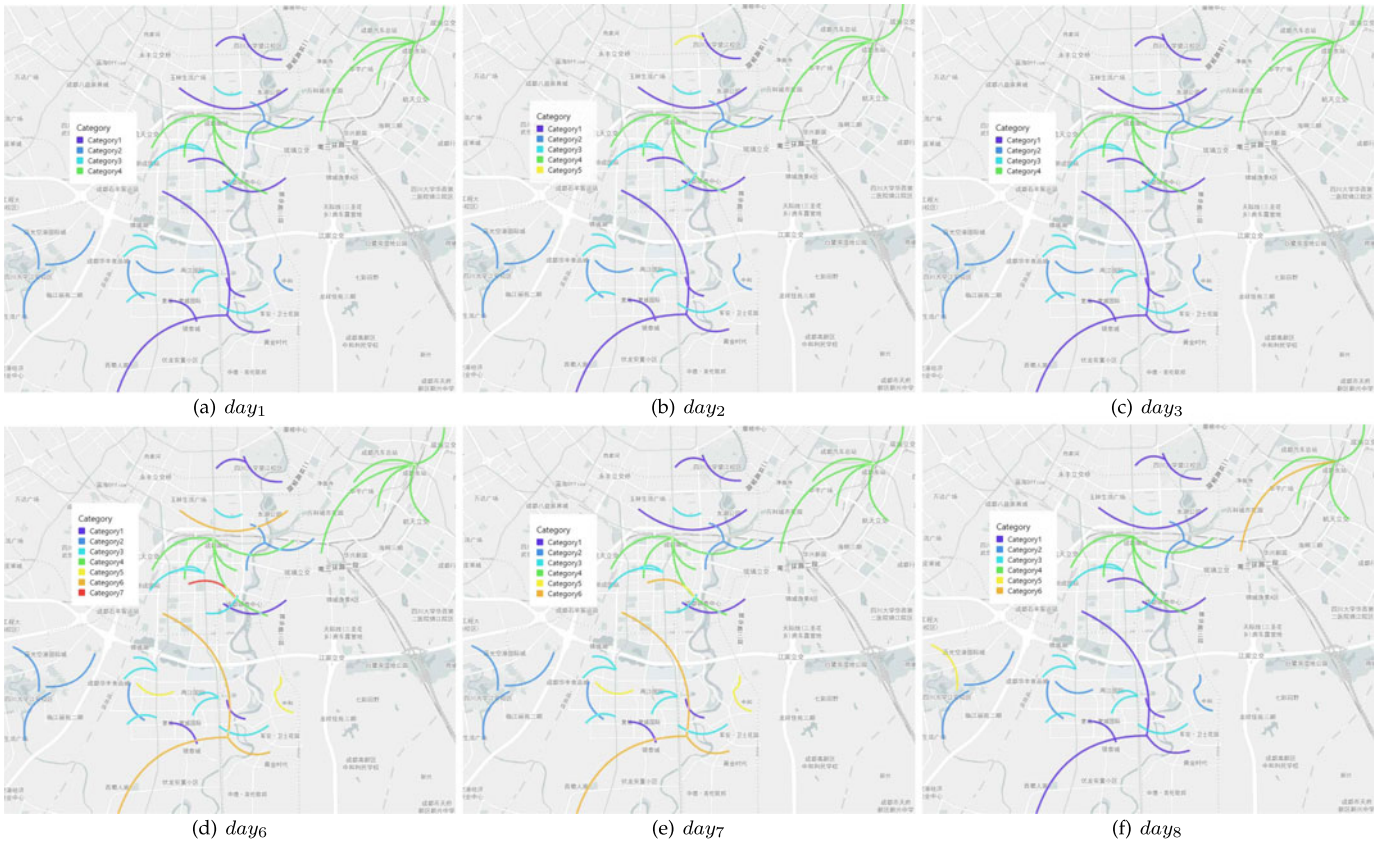
After clustering knowledge according to R, we analyze some knowledge in the same category. We select ten knowledge triplets from four categories on day_1 to display, as shown in Fig. 11a. Fig. 11 illustrates some examples of different categories of regional flow patterns, and Table 5 presents the similarity scores for these categories. Category 4 has the same region pairs (knowledges) on various days and has an average similarity score of 0.9192. This means that the stability of the flow trend in category 4 is better than that of other categories. Relatively, some region pairs



(a) Vector visualization of R (b) Vector visualization of the top 20 r that most similar to r_{39768}

Fig. 9. Vector visualization of 40,000 relations.

Fig. 10. The top five r's that are most similar to r_{sample}.



(a) day_1 (b) day_2 (c) day_3

(d) day_6 (e) day_7 (f) day_8

Fig. 11. Examples of categories for regional flow patterns. day_i expresses week i, for example, day_1 and day_6 mean Monday and Saturday respectively. day_8 represents the Monday of the next week. day_1, day_2, and day_3 show the traffic pattern on weekdays, day_6 and day_7 show the traffic pattern on weekends, and day_8 demonstrates the periodicity of the traffic pattern.

in category 1 changed to category 6 on day_6 and day_7, and the similarity scores on days day_6 and day_7 differed significantly from the other days, indicating that category 1 is susceptible to weekend effects and appears unstable. Similarly, category 2 is also affected by weekends. In particularly, the region pairs in category 3 are consistent on different days, but the similarity scores on day_6 and day_7 are slightly lower than on other days, implying that the flow of category 4 is more variable on weekends compared to weekdays. Moreover, As can be seen from Figs. 11a, 11b, 11c, and 11f, a regional flow pattern r in category 5 on day_2 has changed. According to r belonging to category 1 in other figures, we can consider that the flow pattern between the two regions on day_2 has changed, that is, there may be some abnormal conditions between the two regions, such as traffic accidents causing the flow pattern to occur change.

TABLE 5
Similarity Scores for the Categories of Regional Flow Patterns

Categories	day_1	day_2	day_3	day_6	day_7	day_8	Average
Category1	0.8854	0.8901	0.8748	0.9145	0.9232	0.8793	0.8945
Category2	0.8324	0.8374	0.8297	0.8572	0.8501	0.8431	0.8416
Category3	0.8711	0.8682	0.8641	0.8532	0.8498	0.8763	0.8637
Category4	0.9182	0.9213	0.9119	0.9241	0.9223	0.9177	0.9192
Category5	-	-	-	0.8935	0.8935	-	-
Category6	-	-	-	0.9145	0.9172	-	-
Category7	-	-	-	-	-	-	-

(Sim_C)

Classification Application of V. The more similar the embedding of the regions, the more similar the traffic flow patterns involved in these regions. To classify the similar regions into a category, we randomly select v_1 and v_{50} as labels and calculate the similarity values of the labeled regions and all regions by Equation (6). Then we arrange these regions in descending order according to the similarity values and select the top 9 regions with the largest similarity values. Finally, these 9 regions and labeled regions are classified into one category. The similarity scores for the categories labelled with v_1 and v_{50} are calculated separately, following Equation (6). We use similarity scores Sim_{C-v_1} and $Sim_{C-v_{50}}$ as benchmarks and analyse the results of the classification. The results of region classification are shown in Fig. 12. From the classification results, we can draw these conclusions. First, $Sim_{v_{106-107}}$, which indicates a similarity score between v_{106} and v_{107}, is 0.4961, and $Sim_{v_{168-169}}$ is 0.5728, which are much lower than Sim_{C-v_1} and $Sim_{C-v_{50}}$. This shows that the traffic flow patterns involved in the adjacent regions may be inconsistent and belong to different categories. Second, the similar regions are distributed in various parts of the city, rather than gathered together, such as regions v_{102} and v_{106} with a $Sim_{v_{102-106}}$ of 0.8687, and regions v_{107} and v_{109} with a $Sim_{v_{107-109}}$ of 0.8946. By classifying regions, we can study some practical applications as follows. First, commercial siting application. For merchants, if the business of a restaurant in v_{102} is good, we can open a branch of this restaurant in v_{106} based on $Sim_{v_{102-106}}$ of 0.8687. Second, urban planning application. The functional regions of the city can be divided according to the categories of regions.

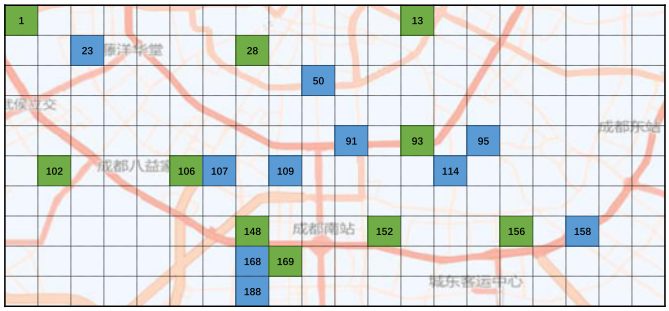


Fig. 12. Region classification with v_1 and v_{50} as labels. The same color represents the same category. Sim_{C-v_1} is 0.8836 and $Sim_{C-v_{50}}$ is 0.8955.

In addition to the above analysis and applications, the knowledge graph of regional flow pattern has many other applications in practice, such as the display of the city's global flow change distribution, personalized route and travel time recommendations.

6 Conclusion

In this paper, we proposed a knowledge mining network for regional flow pattern to mine knowledge of urban flow pattern. First, the features of the regional flow pattern and the region features were extracted as R and V of the knowledge graph. Then, regional POI features were modeled to enhance the embedding representation of R and V. Finally, based on the translation distance method, a RFP-KMN model was proposed to realize the construction of the knowledge triplet of regional flow pattern.

In the future, we can enhance the representation and construction of the regional flow knowledge graph from the following aspects. First, more factors should be considered, e.g. terrain, holidays, accidents, and weather. Then, the optimal granularity of the time period is selected. The regional flow pattern is the change in flow over a period of time. How to choose the most suitable time period is very important to show the best regional flow pattern. On the one hand, the characteristics of regional flow patterns are difficult to extract if the time period granularity is large. On the other hand, if the granularity of the time period is small, the excavated regional flow patterns are meaningless. Finally, the constructed knowledge graph of regional flow pattern should be applied to specific city applications to solve the problems in the city.

References

[1] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–55, 2014.

[2] B. Du *et al.*, "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 972–985, Mar. 2020.

[3] 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 *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 1720–1730.

[4] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, "Urban flow prediction from spatiotemporal data using machine learning: A survey," *Inf. Fusion*, vol. 59, pp. 1–12, 2020.

[5] S. Wang, J. Cao, H. Chen, H. Peng, and Z. Huang, "SeqST-GAN: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 4, pp. 1–24, 2020.

[6] S. Wang, H. Miao, H. Chen, and Z. Huang, "Multi-task adversarial spatial-temporal networks for crowd flow prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1555–1564.

[7] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proc. 24th SIGKDD Conf. Knowl. Discovery Data Mining*, 2018, pp. 965–973.

[8] H. Assem, S. Ghariba, G. Makrai, P. Johnston, L. Gill, and F. Pilla, "Urban water flow and water level prediction based on deep learning, in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 317–329.

[9] B. Wang *et al.*, "Deep uncertainty quantification: A machine learning approach for weather forecasting, in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2087–2095.

[10] S. Berkhahn, L. Fuchs, and I. Neuweiler, "An ensemble neural network model for real-time prediction of urban floods," *J. Hydrol.*, vol. 575, pp. 743–754, 2019.

[11] K. Saleh, M. Hossny, and S. Nahavandi, "Spatio-temporal densenet for real-time intent prediction of pedestrians in urban traffic environments," *Neurocomputing*, vol. 386, pp. 317–324, 2020.

[12] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction, in *Proc. 31th AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[13] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, 2018.

[14] J. Liu, T. Li, P. Xie, S. Du, F. Teng, and X. Yang, "Urban big data fusion based on deep learning: An overview," *Inf. Fusion*, vol. 53, pp. 123–133, 2020.

[15] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[16] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 1–23.

[17] X. Wang, F. D. Salim, Y. Ren, and P. Koniusz, "Relation embedding for personalised poi recommendation," *Adv. Knowl. Discov. Data Mining*, vol. 12084, pp. 53–54, 2020.

[18] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[19] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.

[20] H. Xiao, M. Huang, and X. Zhu, "TransG: A generative model for knowledge graph embedding," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 2316–2325.

[21] C. Zhuang, N. J. Yuan, R. Song, X. Xie, and Q. Ma, "Understanding people lifestyles: cconstruction of urban movement knowledge graph from GPS trajectory," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3616–3623.

[22] A. Padia, K. Kalpakis, F. Ferraro, and T. Finin, "Knowledge graph fact prediction via knowledge-enriched tensor factorization," *J. Web Semantics*, vol. 59, 2019, Art. no. 100497.

[23] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1811–1818.

[24] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proc. 16th Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technologies*, 2018, pp. 327–333.

[25] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A capsule network-based embedding model for knowledge graph completion and search personalization," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 2180–2189.

[26] Y. Xie, H. Xu, J. Li, C. Yang, and K. Gao, "Heterogeneous graph neural networks for noisy few-shot relation classification," *Knowl.-Based Syst.*, 2020, Art. no. 105548.

[27] Z. Xu *et al.*, "Building knowledge base of urban emergency events based on crowdsourcing of social media," *Concurrency Comput.: Pract. Experience*, vol. 28, no. 15, pp. 4038–4052, 2016.

[28] L. Zhao *et al.*, "Urban multi-source spatio-temporal data analysis aware knowledge graph embedding," *Symmetry*, vol. 12, no. 2, 2020, Art. no. 199.

[29] K. Yang and J. Zhu, "Next POI recommendation via graph embedding representation from H-deepwalk on hybrid network," *IEEE Access*, vol. 7, pp. 171105–171113, 2019.
[30] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, "Spatiotemporal representation learning for translation-based poi recommendation," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–24, 2019.
[31] Y. Yu, H. Wang, and Z. Li, "Inferring mobility relationship via graph embedding," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–21, 2018.
[32] Y. Wang, H. Sun, Y. Zhao, W. Zhou, and S. Zhu, "A heterogeneous graph embedding framework for location-based social network analysis in smart cities," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2747–2755, Apr. 2020.
[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[34] Y. Kim, A. Rush, L. Yu, A. Kuncoro, C. Dyer, and G. Melis, "Unsupervised recurrent neural network grammars," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 1105–1117.
[35] G. K. Anumanchipalli, J. Chartier, and E. F. Chang, "Speech synthesis from neural decoding of spoken sentences," *Nature*, vol. 568, no. 7753, pp. 493–498, 2019.
[36] R. Jiang *et al.*, "Deep roi-based modeling for urban human mobility prediction," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technologies Archive*, vol. 2, no. 1, pp. 1–29, 2018.
[37] B. Wang, Z. Yan, J. Lu, G. Zhang, and T. Li, "Deep multi-task learning for air quality prediction," in *Proc. Int. Conf. Neural Inf. Process.*, 2018, pp. 93–103.
[38] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose-invariant embedding for deep person re-identification," *IEEE Trans. Image Processi.*, vol. 28, no. 9, pp. 4500–4509, Sep. 2019.
[39] T. Ying, W. Shining, and L. Wanxiang, "Human ear recognition based on deep convolutional neural network," in *Proc. Chin. Control Decis. Conf.*, 2018, pp. 1830–1835.
[40] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[41] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 301–306.
[42] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[43] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.
[44] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–14.
[45] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.

Jia Liu received the BS and MS degrees from Xihua University, Chengdu, China, in 2018. He is currently working toward the PhD degree at Southwest Jiaotong University. His reseach interests include data mining, machine learning, and knowledge graph, etc.

Tianrui Li (Senior Member, IEEE) received the BS, MS, and PhD degrees from the Southwest Jiaotong University, Chengdu, China, in 1992, 1995, and 2002, respectively. He was a postdoctoral researcher with SCKCEN, Belgium from 2005 to 2006, and a visiting professor with Hasselt University, Belgium, in 2008, the University of Technology, Sydney, Australia, in 2009, and the University of Regina, Canada, 2014. He is currently a professor and the director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or coauthored more than 300 research papers in refereed journals and conferences. His research interests include big data, cloud computing, data mining, granular computing, and rough sets. He is a fellow of IRSS and senior member of ACM.

Shenggong Ji received the PhD degree from Southwest Jiaotong University, Chengdu, China in 2020. He is currently working with Tencent Inc., Shenzhen, China. His major research interests include data mining, reinforcement learning, and urban computing.

Peng Xie received the BS degree from the Chengdu University of TCM, in 2017. He is currently working toward the PhD degree at Southwest Jiaotong University. His reseach interests include deep learning, knowledge graph, and urban computing, etc.

Shengdong Du received the PhD degree from Southwest Jiaotong University, Chengdu, China, in 2020. He is now an associate professor with the school of computing and artificial intelligence, southwest jiaotong university. His research interests include data mining and machine learning.

Fei Teng received the BS degree in communications engineering from Southwest Jiaotong University, in 2006, and the PhD degree in computer science from Ecole Centrale Paris, France, in 2011. She is now an associate professor with the school of computing and artificial intelligence, southwest jiaotong university. Her research interests include cloud computing, optimization of distributed systems and Industrial big data mining. She has authored tens of high quality journal and conference papers, and has been a referee for a number of international journals, including *Information Sciences, Journal of Parallel and Distributed Computing, The Journal of Supercomputing*. She has served as CBPM2013 steering chair and Greencom2015 workshop chair. Now she serves as the academic secretary of CCF Yocsef Chengdu section.

Junbo Zhang (Member, IEEE) is a senior researcher with JD Intelligent Cities Research and the head of AI Department, JD Intelligent Cities Business Unit, JD Digits. Prior to that, he was a researcher with MSRA from 2015–2018. His research interests include urban computing, machine learning, data mining, and big data analytics. He currently serves as associate editor with *ACM Transactions on Intelligent Systems and Technology*. He has published more than 30 research papers (e.g. AI Journal, *IEEE Transactions on Knowledge and Data Engineering*, KDD, AAAI, IJCAI) in refereed journals and conferences, among which one paper was selected as the ESI Hot Paper, three as the ESI Highly Cited Paper. He received the ACM Chengdu Doctoral Dissertation Award in 2016, the Chinese Association for Artificial Intelligence (CAAI) Excellent Doctoral Dissertation Nomination Award in 2016, the Si Shi Yang Hua Medal (Top 1/1000) of SWJTU in 2012, and the Outstanding PhD Graduate of Sichuan Province in 2013. He is a member of ACM, CAAI, and China Computer Federation.