Attention-Based Sequence Learning Model for Travel Time Estimation

ZHONG WANG\textsuperscript{1}, HAO FU\textsuperscript{2}, GUIQUAN LIU\textsuperscript{2}, AND XIANWEI MENG\textsuperscript{2}, (Member, IEEE)

\textsuperscript{1}School of Data Science, University of Science and Technology of China, Hefei 230026, China
\textsuperscript{2}School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China

Corresponding authors: Guiquan Liu (gqliu@ustc.edu.cn) and Xianwei Meng (mxianwei@163.com)

This work was supported in part by the Anhui Sun Create Electronics Company Ltd. under Grant KD1809300321, in part by the National Key Research and Development Program of China under Grant 2018YFC0832101, and in part by the National Key Research and Development Program of China under Grant 2018YFB1801105.

ABSTRACT Travel time estimation (TTE) on a specific route is a challenging task since the complex road network structure and hard-captured temporal patterns. Many excellent methods have been proposed to address the aforementioned problems. Some approaches well designed heuristically in a non-learning based way have the advantage of a quick response to the query for travel time estimation. However, these methods are largely affected by the noise of traffic data since they are limited to a single feature. Existing road segment based methods are generally considered intuitive but are not accurate enough for they fail to model complex factors, like delay and direction of intersections. In this paper, we propose a novel attention based sequence learning model for travel time estimation of a path (ASTTE), that not only considers the real-world road network topology as multi-relational data but also refine the problem to the road segment and intersection direction aspects. Besides, we integrate the traffic information as local and neighbor dependency, which helps to monitor dynamic traffic conditions during the trip. The use of the attention mechanism allows the model to focus on significant elements among the path comprises road segments and intersections. High-quality experiments on two real-world datasets have demonstrated the effectiveness and robustness of our framework.

INDEX TERMS Travel time estimation, road network topology, multi-relational data.

I. INTRODUCTION

Travel time estimation for a given route is a kernel and foundation component of the intelligent transportation system (ITS). With the acceleration of the pace of life and a strong sense of time among the contemporary, the increasing challenges are obvious in the area of travel time estimation (TTE) tasks. Accurate travel time estimation provides a reference for people to make arrangements in advance and also helps drivers to choose the appropriate route and arrive at the destination in time [1], [2], thus effectively avoiding aggravating traffic congestion [3], [4]. Furthermore, the construction of the navigation system like global positioning system (GPS) and equivalent sensor facilitates collecting abundant trajectory data that interest many scholars and arouse their research enthusiasm on the transportation area.

The strategically essential task TTE has been widely studied in the past, from the early stages of research, some explorations [5]–[7] utilize historical travel time of the data which consists of millions of tracks with rich time-series features, then TTE is decomposed into searching for path and multivariate time-series prediction task. In [8], the authors leverage a weighted average of the trips geographically adjacent to the query path in history and combine it with other related variables to estimate the travel time. It is widely used in these methods dividing the path into road segments and intersections, if we glance over the picture shown in Fig. 1(a) which depicts a track on the real road network, the overall travel time of a trip given the route can be defined as the summation of consuming time on passing roads and delay time of intersections [9]. However, the picture in Fig. 1(b) represents the detailed structure of an intersection the path involved, the travel delay of car A and car B is obviously different for the effect of traffic lights and the traffic condition of the next road segment even they stand in the same
intersection. Besides, it is intuitive to understand the time crossing the road segments makes up a large part of the overall travel time of a path in most cases, which is the reason that intersections always with less discussion in many works [10], [11]. Nevertheless, congestion usually happens in the surrounding of intersections during rush hours. How to explore non-linear and complex sequence data to discover its inherent time-varying consuming patterns and make accurate travel time prediction is a challenging task.

Recent years have witnessed the inspiring potential of deep neural networks for its powerful ability to automatically extract features from massive data. Many advanced algorithms are proposed in the task of travel time estimation to learn the complex road network structure using deep neural networks. The authors design a multi-task representation learning model to learn spatial prior knowledge of the topological structure, and the objective function is to predict the travel time as well as other auxiliary tasks [12], the increasing additional information like the number of traffic lights has devoted to making more accurate travel time estimation. [13] partitions the whole road network into some disjoint but equal-sized grids and adopts the distributed representation to represent each grid. Nevertheless, modeling the topology to a regular graph is inconsistent with the real road network structure. Also, travel time estimation suffers from more uncertainties since it is a long-lasting behavior. For instance, it always happens in realistic scenarios that a forty minutes ride-hailing trip starts at 5:00 PM will drive with a relatively stable speed at first and slow down as evening peak hour traffic floods into the city. It is of great significance to adaptively learn the information involved in the route to provide dynamic traffic conditions during the trip.

To tackle the aforementioned challenges, we propose an attention based deep sequence learning model for travel time estimation of a path (ASTTE). Specially, we propose a novel underlying road network embedding method to obtain a unified representation space for learning latent vectors, which could capture the relationships of adjacent road segments among different intersection directions. Different from the traditional embedding methods which are based on regular grids or cursory graphs, our neural embedding method generates road segments and intersection directions representations through exploiting the real-world road network with taking road restrictions into consideration. Then, with the underlying topology embedded, we design a road segments field and an intersection directions field combine with traffic information to model trip present consumption patterns from historical sequential data. We can summarize the major contributions of our work as follows:

- We present a new perspective to model the real-world road network topology as multi-relational data and propose an improved method to interpret the intersection directions as translations operating on the low-dimensional embeddings of the road segments.
- We utilize the graph convolution neural network to embed the traffic condition and integrate it both as local and neighbor dependency.
- We apply the attention mechanism to focus on significant elements of the path from the sequence data, which helps to learn time-varying consuming patterns.
- We construct experiments on two large-scale datasets to evaluate our proposed model. The empirical results demonstrate the effectiveness and robustness of our framework are better than advanced baselines.

II. RELATED WORK

A. TRAVEL TIME ESTIMATION

The forecast of route travel time has been studied extensively, which aims at computing the usage time on a given path. The methods of travel time estimation can be classified into path-free models, global models, and simple additive models. Path-free methods aim at predicting the travel time without the route of the trip and the available information is only limited to origin location, destination location about the path. A neighbor-based approach TEMP proposed in [8], which looks for the trip with near origin and destination from millions of trajectory data, and estimates time duration based on the historical results. To compensate for the deficiency of depending only on a single information source in TEMP, ST-NN [14] investigates the use of distance and time information to estimate travel time. However, the method depends on complicated feature engineering and lack of the ability to model dynamic traffic conditions. Path-free methods attempt to simplify the process, which causes them to be ineffective...
for some occasions that require high accuracy since the limited information.

Global models formulate travel time estimation as a regression model. The conventional regression model support vector regression (SVR) is utilized in [6] to predict travel time for highway users and performs well for traffic data analysis. Long Short-Term Memory (LSTM) network [5] constructs a deep structure in terms of time to produce high accuracy travel time prediction. [15] maps the historical GPS trajectory into a higher dimensional feature space from the lower input space and uses a deep neural network (DNN) to capture spatial correlations for travel time estimation. However, these methods are discussed without considering the structure of the urban road network and do not apply to the input is a path without GPS points.

Simple additive models are based on the physical structure of the road network. The statistical model [9] combines a spatial moving average (SMA) structure to use vehicle trajectories obtained from low frequency GPS probes and calculates the overall travel time by modeling the segments in a path. To work out the error accumulation in simple additive models, [7] builds a 3D tensor which contains driver, road segment, the time slot to predict the future travel time of each road segment. However, the simple rules used in additive models can not model detailed traffic conditions such as intersection directions and will cause an inaccurate result.

B. REPRESENT LEARNING

Represent learning is a collection of technologies that convert the raw data into features that can be applied effectively by models. Before represent learning appears, this work is completed by manual feature learning which is time-consuming and requires professional knowledge. Represent learning is designed to study the underlying structure of the data, therefore, it can extract and analyze the latent relationships between data.

Represent learning is growing fast and has been widely used in artificial intelligence and machine learning, such as natural language processing (NLP) [16], knowledge graph (KG) [17], and graph neural network (GNN) [18]. Word2vec model is a successful application of represent learning in natural language processing, it maps each word to a vector to represent the relationship between words [16]. Later, the proposed DeepWalk [19] which uses local information obtained from truncated random walks successfully solves the problem of modeling complex relationships in the graph structure. The authors propose TransE [17] to consider the problem of describing various entities and concepts that exist in the real-world and is a canonical model for it is easy to train.

The determination of efficient paths in the transportation network is the important basis of travel time, so there is a lot of work to learn the representation of the underlying road network. In existing work, it is common to partition the entire road network into some disjoint but equal-sized grids, and the vehicle trajectory can be represented as a sequence of grids. However, modeling the topology to a regular graph may cause a loss of import information like intersections. Recently, MURAT [12] has been proposed to model the road network structure to an undirected graph which consists of road segments and intersections. The key difference between our model and MURAT processing road network information lies in that we propose to use the TransE to learn a deeper representation for each road segment and discuss more details of intersections.

III. PRELIMINARIES

Definition 1 (Road Network): A road network can be represented as an undirected graph of roads $G = (V, A)$, where vertex $v \in V$ denotes each road segment and $N = |V|$ is link numbers. $A \in R^{N \times N}$ denotes the connectivity among links, $A_{i,j}$ is 1 if road $i$ and road $j$ are directly connected through an intersection. Fig. 2 is the road network after the transformation of the area shown in Fig. 1(a), the links marked in green represent a movement track of the vehicle.

Definition 2 (Origin GPS Data): $T = (g_1, g_2, \ldots, g_n)$ is a sequence of GPS records of a trip where each record $g_i$ is specified as $(g_i, \text{long}, g_i, \text{lat}, g_i, \text{time})$ tuple of the location. With the help of map matching [20], we can easily match it to a specific road segment.

Definition 3 (Mapping Data): After map matching, a trip is represented as $T = (l, s, e)$, where $l = \{l_i\}$ is transformed path sequence represents the trajectory of the trip, $i \in [1, N]$, $|l| = n$, $s$ and $e$ denote the departure time and arrival time, respectively. The movement path involved in $n$ road segments and the actual travel time is given by $e - s$.

Definition 4 (Problem Statement): Given a trajectory data contains a path sequence $l = \{l_i\}$, $i \in [1, N]$, $|l| = n$, the departure time $s$, and the additional traffic condition information $X^r = (X^r_1, X^r_2, \ldots, X^r_N) \in R^{N \times f_i}$, where $X_i^r$ denotes the signal of node $i$ at $r$-th time interval in the graph $G$ referred before, $f_i$ represents the dimension space. Our task is to estimate the travel time of the trip.

IV. PROPOSED FRAMEWORK

Fig. 3 shows the overall architecture of the novel framework attention based deep sequence learning model for travel
time estimation of a path (ASTTE). First, we represent the approach to model the underlying road network structure as multi-relational data refer to undirected graphs whose nodes correspond to road segments and edges indicate the intersection in Translating Embedding of Road Network component. After getting the embedding of the vehicle movement track, we show how to calculate the time travel through the road segments and travel delay during the turns, respectively. Finally, we apply the attention mechanism combines with meaningful attribute information to integrate and focus on significant elements of the path.

**A. TRANSLATING EMBEDDING FOR ROAD NETWORK**

In the first stage of ASTTE, Translating Embedding for Road Network aims at generating unified representation for underlying road network structure by interpreting intersection directions as translations operating on the low-dimensional embeddings of the road segments. Previous works [11], [13] always split the area into equal-sized grids to represent spatial embeddings, and successfully learn a variety of feature information of the grids. However, expressing the road network with a regular grid fails to reproduce the real-world topological structure. In a real-world road network, there exists a lot of limitations for enforcing driving behaviors, such as roadway classification (one-way, two-way), number of lines, and turn restrictions at junctions. As the picture shows in Fig. 1(b), car A turns its left for the only dependency between current and left positions, which similar to the one-to-one relations in the knowledge graph domain. For instance, from the well-known fact “Barack Obama was born in Hawaii.”, an easily constructed triple (Barack Obama, place of birth, Hawaii) contains entities and relationships. Motivated by this, we consider the road segments as multi-relational data and the intersection directions are relationships between them. TransE [17] is one of the promising methods with negative sampling to embed entities and relations into a continuous vector space to learn embeddings of the knowledge base. We creatively introduce it to represent the real-world road network. Besides, instead of merely focusing on the direct relations in TransE, whether there is a second-order path between negative samples is also noticed.

In this paper, we propose an improved TransE to generate road segments and directions representations by the means of exploiting the connectivity between road segments. Specifically, we traverse the adjacency matrix $A$ of graph $G$, if $A_{ij} = 1$, then create triple $(r_i, d_i, r_j)$, where $d_i$ indicates that there exists a direction of intersection from road segment $r_i$ to $r_j$. Noted that the relationship from road segment $r_j$ to $r_i$ is inconsistent despite passing the same intersection, which means it consists of another triple $(r_j, d_k, r_i)$ and it is quite in accord with the actual situation. Then, we can easily obtain a set $S_p$ contains all positive triples and $|S_p| = M$.

Different from the only focus on direct relations in TransE, we have also considered the spatial dependencies between road segments connected with the same intersection through different directions. Along this line, as shown in Fig. 4,
we add a tolerance coefficient to negative sampling and propose a novel road network embedding method, called t-TransE. The t-TransE learns road segments embedding \( \overrightarrow{R} = (\overrightarrow{r_1}, \overrightarrow{r_2}, \ldots, \overrightarrow{r_N}) \in \mathbb{R}^{N \times f^2} \) and intersection directions embedding \( \overrightarrow{D} = (\overrightarrow{d_1}, \overrightarrow{d_2}, \ldots, \overrightarrow{d_M}) \in \mathbb{R}^{M \times f^2} \) in the follow way: \( \overrightarrow{r_i} + \overrightarrow{d_l} \approx \overrightarrow{r_j} \), when the triplet \((r_i, d_l, r_j)\) hold in \( S_p \), and randomly select a road segment \( r_k \) to replace \( r_i \) or \( r_j \) (not the same time). We suppose \( r_k \) as a substitute for \( r_i \), intuitively, \( \overrightarrow{r_k} + \overrightarrow{d_l} \) should be far away from \( \overrightarrow{r_j} \). However, if the \( A_{kj} \) is 1, which means \( r_i \) and \( r_j \) are at least second-order reachable, we should preserve the similarities of road segments in real-world topological structure. Thus, the objective loss is:

\[
\mathcal{L} = \sum_{(r_i, d_l, r_j) \in S_p} \sum_{(r_k, d_l, r_j) \in S_n} [\alpha + \mathcal{L}_p - \mathcal{L}_n + t \cdot A_{k, j} \cdot L_r]_+
\]

where \( [\cdot]_+ \) represents the hinge loss function, \( \alpha \) is a margin hyper parameter. We select the \( L_2 \) norm as the distance measurement. \( t \) is the tolerance coefficient and \( A_{k, j} \) is 1 if vertex \( r_j \) is connected to \( r_k \). \( S_n \) is the set of negative triplets.

Instead of training with the entire network, we pre-train the t-TransE to learn a representation of road segments and directions based on operating it to the low-dimensional vectors. In this way, road segments geographically adjacent will be closer in the embedding space.

**B. ROAD SEGMENT FIELD**

The time crossing the road segments makes up a large part of the overall travel time of a path. So it is effective to study the correlations between road segments from historical sequence data. Besides, there exists a difference between travel time estimation and other traffic condition parameters like traffic flow forecast with optional prediction time intervals. Travel time estimation suffers from more uncertainties since it is a long-lasting behavior, which causes a dynamically changing temporal dependency. As mentioned before, a trip starts around the rush hours may suffer a change significantly in the traffic condition. For the above reason, it is desirable to adaptively trade-off dependency from traffic information in local and neighbor perspectives. In our paper, traffic information refers to the road segment speed exactly.

1) **LOCAL DEPENDENCY**

For each road segment, the speed of historical time series directly adjacent moment of trip departure inevitably influences the future traffic condition. As proposed in [12], there exists a smooth transition between adjacent time intervals, thus, it is desirable to learn road segments information provided by traffic data.

The traffic condition of each road segment is defined by the average speed traveled by the vehicles and can be expressed as:

\[
\chi_i^\tau = \frac{\sum_{q=1}^{Q} v_i^q \tau}{Q}
\]

where \( \chi_i^\tau \) represents the speed of road segment \( i \) at \( \tau \)-th time interval, \( Q \) is the number of vehicles travel in the road segment \( i \) at \( \tau \)-th time interval and \( v_i^q \) is the speed of vehicle \( q \). Therefore, we obtain the traffic state of each road segment involved in the trip as a local dependency.

2) **NEIGHBOR DEPENDENCY**

Because of the routine of people, the traffic state shows repeated daily and weekly patterns. It is essential to take advantage of the regular patterns in previous traffic data. We can set the \( \tau \)-th time interval to be the time half an hour later of the trip departure, and which helps to give a sense of the future from the traffic conditions at the same time yesterday. Existing work [21] considers the repeated traffic information limited to the road segment level, whereas the calculated speed may not be accurate enough for the sampling rate or GPS error in some trajectory data. It is necessary to learn the neighbor dependency to illustrate the true traffic condition of the entire city. In addition, the specific road segment can be affected by its neighboring road segments [22], therefore, we apply graph convolutional network (GCN) to learn the road network level information as a neighbor dependency.

We adopt the spectral graph theory with graph Fourier transforms introduced in [18], which is often called the spectral graph convolution. As for the graph \( G = (V, A) \) we defined before, \( D \) denotes the diagonal matrix, the symmetric normalized Laplacian can be represented as \( L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \). Then the operation on filtering \( K \)-localized in space defined as:

\[
\hat{X}^\tau = \sigma\left(\sum_{k=0}^{K-1} \theta_k L^k X^\tau\right)
\]

where \( L^k \) is the \( k \)-th power of the graph Laplacian matrix, \( \sigma(\cdot) \) is an activation function and \( \theta_k \) is the trainable coefficient.
Then, we can get the traffic information of each road segment $i$ with neighbor dependency as $\hat{X}^T_i$.

### 3) LONG SHORT TERM MEMORY

Instead of estimating the travel time of each road segment individually in many works [9], [23], recurrent neural network (RNN) is applied to capture the temporal dependencies among road segments. Suppose the trajectory of a taxi order involved in $n$ road segments is represented as a sequence $l = \{r_i\}, i \in [1, N], |l| = n$, after embedding lookup from $\mathcal{R}$, we get the vector sequence $\pi = \{\tau_i\}$, then we concatenate the traffic state to the road segment vector.

$$\hat{r}_i = \tau_i \oplus X^T_i \oplus \hat{X}^T_i$$  

(7)

the concatenation operator is denoted as $\oplus$ and transformed sequence $s_i$ can be represented as $s_i = \{\hat{r}_i\}, i \in [1, N], |s_i| = n$.

Recurrent Neural Network (RNN) [24] architecture contains a feedback loop that enables the information in the previous step could be leverage at the current step of sequential input. Long Short Term Memory (LSTM) [25] is a variant of RNN and alleviates the problem of vanishing gradient by adding gate structures and memory cells. We feed the sequence $s_i$ to LSTM, which not only integrates dynamic traffic information but also can learn the inherent regular temporal patterns. Formulaically:

$$i^T = \sigma(W_i^1 s^T + W_i^2 h^{T-1} + b_i)$$  

(8)

$$f^T = \sigma(W_f^1 s^T + W_f^2 h^{T-1} + b_f)$$  

(9)

$$o^T = \sigma(W_o^1 s^T + W_o^2 h^{T-1} + b_o)$$  

(10)

$$c^T = f^T c^{T-1} + i^T \cdot \tanh(W_c^1 s^T + W_c^2 h^{T-1} + b_c)$$  

(11)

$$h^T = o^T \cdot \tanh(c^T)$$  

(12)

where $i^T$, $f^T$, $o^T$ represent input, forget, and output gates at $\tau$-th time interval respectively. $W$ and $b$ are trainable parameters, and $\sigma(\cdot)$ denotes activation function.

### C. INTERSECTION DIRECTION FIELD

From the trajectory sequence $l$ consists of road segments, we can easily match the intersection direction sequence $z = \{d_j\}, j \in [1, M], |z| = m$ with the help of set $S_p$ we construct in Translating Embedding for Road Network component. Noted that the noise in GPS positioning data is inevitable, which may cause the adjacent road segment in $l$ whereas not connected in the real road network. In this condition, we overlook it and add nothing to direction sequence $z$.

It is common practice to ignore the time crossing the intersection in existing works [10], [11]. However, the delay time on the intersections is just as important for there are always a lot of cars up at the crossroads, in particular, the peak commuting, which causes the overall travel time of a trip highly depends on some key intersections involved. Ignoring or simply mixing the intersections may lose some critical information. Therefore, first, we transform $z$ to the intersection direction vector sequence $\overline{z} = \{\overline{d}_j\}$ through the embedding lookup from $\overline{D}$. Next, we utilize the convolutional layer (1D-CNN) [26] to learn the local spatial relationships between adjacent intersection directions to extract the salient local features to represent the intersections. A feature $c_i$ is generated from intersection directions $\overline{\tau}_{ci+p}$ with a window $W_{\text{conv}} \in \mathbb{R}^{p \times f_2}$.

$$c_i = \sigma(W_{\text{conv}} \overline{\tau}_{ci+p} + b_{\text{conv}})$$  

(13)

$W_{\text{conv}}$ and $b_{\text{conv}}$ are trainable parameters, and $\sigma(\cdot)$ denotes activation function. After a convolution operation on $\overline{\tau}$, we can get a feature map $C = (c_1, c_2, \ldots, c_{m-p+1})$. Our model uses multiple filters $f_3$ to obtain $s_d \in \mathbb{R}^{f_3 \times (m-p+1)}$ as the output of the Intersection Direction Field component.

### D. TRAVEL TIME ESTIMATION

The inherent time-varying consuming patterns make it hard for travel time estimation. Learning and exploiting the implicit factors is effective and interpretable, therefore, we extract some attribute data on each trip to help the prediction. Finally, we apply the attention mechanism to balance trajectory information mixed with attribute data.

1) ATTRIBUTE DATA

We select attributes of the following types: peak-hour flag, temporal-spatial information, and driver behavior. Peak-hour refers to the period that normally happens twice every day and means more traffic jams on the road especially on weekdays. As aforementioned, congestion always appears around intersections, it is beneficial to send the lighted message into model learning. In our work, we divide a day by the hour, and define the morning peak from 7:00 to 9:00 AM and evening peak from 5:00 to 7:00 PM and the other time of day is the non-peak hour. Temporal dependency consists of daily and weekly information, it’s straightforward how different between weekdays and weekends. For the daily information aspect, however, the dependencies between road segments could change over time. For example, people go to the manufacturing district from the residential area in the morning and come back in the evening, which causes a time-varying impact on intersection directions though cross the same road segments. The behavior of the driver is related to the traffic flow on the road network, which would innately influence the travel time. For instance, drivers who are considered aggressive are those who travel at relative speeds above average, even tend to jump yellow lights on busy crossroads at full speed, which would make shorter travel time to the destination. Since we include this factor in our attribute representation, we implicitly take the behavior of each driver into consideration as well.

Despite the deep learning technology has shown its power in many fields [16], [27], [28], the nature of our raw attribute data makes it hard to lend itself to be input into the model directly. For categorical features like driver information, similar to [29], we use an embedding layer to effectively reduce the feature dimension and map sparse features to dense representations. Normalization is the formal process of analyzing
the continuous data, which avoids the special process for the exception. A continuous feature like distance $x$ is transformed to $\tau$ by formula $\tau = \frac{x - \mu}{\sigma}$, where $\mu$ denotes the mean value and $\sigma$ denotes the standard deviation of $x$. To supercharge the model learning with non-linearities, we also input the powers $\tau^2$ and a root $\sqrt{\tau}$. Finally, we concatenate all attribute vectors as $E_a$.

2) ATTENTION MECHANISM

The attention mechanism is first applied in computer vision [30], which is used as personal perceptual filters to limit detection in the smaller area. The kernel of attention mechanism is focusing limited attention on import information. Recall that we obtain the hidden sequence $H = (h^1, h^2, \ldots, h^n)$ in the road segment field component, and $s_d$ represents the key features vectors of intersections in the intersection direction field component. For the integration of a taxi order trajectory information, we use $h$ to represent the result of appending the $s_d$ to $H$. The assumption that different road segments and intersection directions contribute equally to the total travel time, which does not reflect reality. For instance, vehicles tend to consume time driving in the congested roads and junctions at peak hours, whereas paying more time to longer roads when the traffic is smooth. Therefore, the weights of elements in $h$ need to be reallocated combine with attribute information $E_a$.

$$d^h_i = \frac{\exp(\hat{h}_i W_h E_a)}{\sum_{i=1}^{n+m-p+1} \exp(\hat{h}_i W_h E_a)}$$ (14)

$$U^h = \sum_{i=1}^{n+m-p+1} d^h_i \hat{h}_i$$ (15)

where $W_h$ has the corresponding shape. Then, we feed the $U^h$ to feedforward neural network to make a final prediction result $y$. The pseudocode of the algorithm is shown in algorithm 1, which presents an overview of the learning process.

V. EXPERIMENT

A. DATASETS AND DATA PROCESSING

We construct a general framework to evaluate our proposed model in two real-world large-scale datasets in Hefei and Chengdu respectively. Hefei is the capital of Anhui Province, China. It covers an area of about 114,451 square kilometers and the research area in our work is within the second ring road. The ride-hailing order data of Hefei Dataset are collected from 1st Nov. to 25th Dec. in 2019. Table 1 shows an example of a trip travels in the path of Hefei and its trajectory described in Fig. 1(a). When the value of the vehicle state is taken as 0 for no passengers aboard and 1 for having passengers. All the data are derived from Didila, which is the biggest online car-hailing company in China. We use the data in the first 40 days for training and the rest for testing. Chengdu Dataset is an open data and could be available at the platform DataCastle. The dataset including trajectories of 14864 taxis in August 2014 in Chengdu, China. The data of the first three weeks are used as the training set and the last 7 days as the test set. Considering the influence of data sparsity, the competition platform eliminates the data from 00:00 to 05:59. For an intuitive understanding of the data distribution, Table 2 shows the statistical results of the two datasets, here, stddev represents the standard deviation most commonly used in probability statistics as a measure of the dispersion of a dataset of values.

The original record data can not be fed into the model directly to obtain the travel time. The data processing is carried out as follows: first, it is necessary to extract the record of each taxi order according to the field of terminal

| TerminalID    | State | Longitude | Latitude | Timestamp         | Speed(km/h) |
|---------------|-------|-----------|----------|-------------------|-------------|
| 18161100166   | 0     | 117.275174| 31.860184| 2019-11-04 01:05:44| 19.4        |
| 18161100166   | 1     | 117.275957| 31.861207| 2019-11-04 01:05:54| 39.8        |
| 18161100166   | 1     | 117.279234| 31.861708| 2019-11-04 01:06:04| 51.1        |
| ...           | ...   | ...       | ...      | ...               | ...         |
| 18161100166   | 0     | 117.285817| 31.866574| 2019-11-04 01:29:33| 12.9        |

Algorithm 1 Attention Based Sequence Learning Model

**Input:** Road network $G$, traffic data $X$, trajectory sequence $l$, attribute data $x$

**Output:** Travel time prediction $y$

1: procedure ASTTE
2: $\bar{R}, \bar{D}, S_p, S_o \leftarrow t$-TransE($G$)
3: $\hat{X}^t \leftarrow GCN(X^t)$
4: for $i \leftarrow 1, \ldots, len(l)$ do
5: $\tilde{r}_i \leftarrow$ map $r_i$ through embedding lookup from $\bar{R}$
6: $\hat{r}_i \leftarrow$ concatenate($\bar{r}_i, \bar{X}^i_1, \bar{X}^i_2$)
7: $s_r \leftarrow$ add $\tilde{r}_i$ to the transformed sequence
8: $z \leftarrow$ select $d_j$ from $S_p$
9: $\bar{z} \leftarrow$ add $\bar{d}_i$ through embedding lookup from $\bar{D}$
10: $H \leftarrow$ LSTM($s_r$)
11: $s_d \leftarrow$ CNN($\bar{z}$)
12: $E_a \leftarrow$ embed($x$)
13: $\hat{h} \leftarrow$ append $s_d$ to $H$
14: $U^h \leftarrow$attention($\hat{h}, s_d, E_a$)
15: $y \leftarrow$ prediction($U^h$)
16: return $y$

1: https://www.xiaojukeji.com
2: https://www.dejingsai.com/v2/index.html

TABLE 1. Description of the trip on Hefei which trajectory described in Fig. 1(a).
number and vehicle state. Further, given a location consists of longitude and latitude, with the assistance of map matching [20], we can easily match it to a specific road segment. Next, we get the speed information based on Eq. (5) for the Hefei dataset. It is noted that the speed field is not directly available in Chengdu dataset, so we calculate the distance based on the Haversine formula. Specifically, suppose we get two adjacent sampling points $g_i$ and $g_{i+1}$ of a vehicle $q$, which we referred to in Definition 2, then the speed field $v_q^i$ can be obtained as follows:

$$v_q^i = \frac{\text{Haversine}(g_i, \text{long}, g_i, \text{lat}, g_{i+1}, \text{long}, g_{i+1}, \text{lat})}{g_{i+1} \text{time} - g_i \text{time}}$$ (16)

where $g_i, \text{long}, g_i, \text{lat}$, and $g_i, \text{time}$ represent the field longitude, latitude, and timestamp respectively. Among them, function Haversine can be called in Python package haversine to calculate the distance according to latitude and longitude. Finally, we construct the attribute data, for instance, weekly information can be easily obtained from the timestamp field.

### B. BASELINE METHODS

We compare our model with support vector regression (SVR) [6], TEMP [8], and advanced method DeepTTE [15]. SVR is an application of support vector machine on regression problems. The objective function of SVR is to find a hyperplane so that the distance between all data at this hyperplane is the smallest. The features used in SVR contains daily information, weekly information, and distance information. We try three different kernel functions, polynomial kernel, Gaussian kernel, and linear kernel. TEMP is a neighbor-based method for travel time estimation, which aims at predicting the travel time based on the “neighbor” trajectories instead of using route information. DeepTTE estimates the travel time based on the given route of the trip and it achieves state-of-art results on the open Chengdu datasets. We implement these algorithms following the parameter settings suggested in the paper. However, it is worth noting that DeepTTE is an end to end model, and it processes the GPS positioning data directly. If the sampling frequency of the data is uniform, it will cause information leakage, the model can easily make an accurate estimation by counting the number of positioning points. So we random resampling the data.

#### C. PARAMETER SETTINGS

We implement our proposed network with the PyTorch toolbox. During the pre-training t-TransE in Translating Embedding for Road Network component, we set the embedding dim $f_2$ to 128, the margin hyperparameter $\alpha$ to 1 and the tolerance coefficient is 0.1 on Hefei dataset; embedding dim $f_2$ to 64, the margin hyperparameter $\alpha$ to 1 and tolerance coefficient is 0.1 on Chengdu dataset. For all datasets, we just use road speed information, so the dimension space $f_1$ is 1. In the graph convolution operator, we set filtering degree $K = 2$ and the output dimension is 64. The number of neurons in the hidden layer of LSTM and the number of filters $f_3$ of CNN are set to 128. The convolution kernel size is 5. We choose relu as the activation function and adopt adam optimization algorithm to train the parameters. The learning rate of adam is searched in $[0.0001, 0.001, 0.005, 0.01]$ and the batch size during training is 100.

#### D. EXPERIMENT RESULTS

1) EVALUATION METRICS

All the models in this paper are evaluated by using multiple evaluation metrics in our experiments, Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE), which can be expressed as follows:

$$
\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{\hat{y}_i} \right| \tag{17}
$$

$$
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{18}
$$

$$
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{19}
$$

where $y_i$ is the prediction result of sample $i$, and $\hat{y}_i$ represents the ground truth.

2) PERFORMANCE COMPARISON AND RESULT ANALYSIS

Table 3 shows the performance of our proposed model with comparison methods on the two datasets. We observe that the SVR suffers from dealing with complicated relationships between features, with the worst result. The performance has greatly improved by TEMP, however, there is an obvious difference in performance on the two datasets. The factor could be that the underlying road network structure of Hefei selected in our experiment is about 30% larger than that of Chengdu and contains longer trajectories that TEMP can not estimate accurately for lacking neighbor trajectories in historical observations. The collective approach DeepTTE makes good results in both the two datasets among the baselines, which demonstrates the power of deep learning in constructing the features of historical traffic data. Obviously, on the whole test set, the proposed model ASTTE achieves the lowest overall MAPE and two other metrics. Further, in order to describe the effectiveness of t-TransE in detail, we discuss

### TABLE 2. The statistical information of the datasets.

| Dataset       | Hefei                  | Chengdu                |
|---------------|------------------------|------------------------|
| area of study region | 306.0km²               | 225.0km²               |
| longitude stddev | 0.038                  | 0.049                  |
| latitude stddev  | 0.032                  | 0.041                  |
| travel time stddev | 597.62s               | 646.37s               |
| travel time mean  | 1399.24s               | 1555.75s               |
| travel distance stddev | 4.43km                | 3.96km                 |
| travel distance mean   | 10.23km                | 9.57km                 |
| number of drivers   | 10042                  | 14864                  |
strategies to realize our model from two aspects: ASTTE-R and ASTTE-T. In ASTTE-R, the component of Translating Embedding for Road Network is eliminated, moreover, we use the one-hot encoded category features to replace road vector and intersection vector as input to LSTM and CNN, which means the model will not learn the spatial dependency. Even then, our ASTTE-R, achieves a similar result with advanced baseline DeepTTE, proving the advantages of our model in utilizing underlying road network information.

In ASTTE-T, we use the original TransE proposed in [17] to replace t-TransE, specifically, we set the tolerance coefficient $t$ in Eq. (1) to 0 during the pre-training. The difference of results between ASTTE and ASTTE-T shows that it is meaningful to take implicit relationships between road segments into consideration.

3) PERFORMANCE OF MODEL ROBUSTNESS
It is challenging to estimate the travel time when a journey with more uncertainties, such as trips with longer time duration and travel around peak hours. In this section, we carry out the experiments to examine the model sensitivity of trip uncertainties on the Hefei dataset. We select the trips which travel time around 40 minutes as long trips and 20 minutes as short trips for contrast, sifting out the trips with departure time from 6:00 AM to 6:00 PM. Fig. 5 shows the fluctuation of the prediction performance of our model ASTTE and advanced model DeepTTE as the time varying. Compared to the long trip prediction task, the two models show better results on the trips consuming time around 20 minutes. Since DeepTTE lacking information about congestions, it suffers the noise of the traffic data in the morning and evening peak hours. We can see that the fluctuating range of the DeepTTE curve is higher especially at rush hour and it becomes worse when the travel duration becomes longer. Overall, the prediction of our proposed model fits the ground truth well than DeepTTE and it is less affected by the rush hour. In Fig. 6, we show the error rates of the models for the trajectories with different time durations. When the time duration gets longer, SVR and TEMP methods make a very inaccurate result. DeepTTE shows a better result for the travel time fewer than 30 minutes. However, the lack of considering traffic information makes DeepTTE hard to handle the long paths. In contrast, our model significantly outperforms other methods with a trajectory longer or shorter. When constructing attribute data, the temporal information including time of a day and day of a week. The week information makes the model successfully learn the different traffic patterns from weekdays and weekends. A display of model performances show in Fig. 7, obviously, ASTTE outperforms TEMP and DeepTTE across all days. It is interesting to observe that the trend of error rate is similar across the model ASTTE and DeepTTE which both share the week information, the performance of Tuesday, Wednesday, and Thursday are much better and are similar to each other, while estimation errors are higher on Monday and Friday near the weekend which are less regular.

4) EFFECT OF TRAFFIC INFORMATION
Traffic information can reflect real road conditions simply and effectively. In order to explore the impact of traffic information on our model ASTTE, we construct the variant model

---

**TABLE 3.** Performance comparison of different approaches for the travel time estimation on the datasets.

| Method   | Hefei         | Chengdu       |
|----------|---------------|---------------|
|          | MAE(s) | RMSE(s) | MAPE(%) | MAE(s) | RMSE(s) | MAPE(%) |
| SVR      | 256.97  | 406.85  | 59.12   | 247.40  | 361.71  | 52.29   |
| TEMP     | 319.04  | 432.15  | 29.48   | 283.12  | 396.34  | 24.25   |
| DeepTTE  | 213.55  | 327.26  | 19.26   | 188.31  | 294.55  | 18.01   |
| ASTTE-R  | 227.18  | 301.44  | 19.57   | 200.48  | 273.35  | 17.76   |
| ASTTE-T  | 194.77  | 286.12  | 17.82   | 159.18  | 258.19  | 16.22   |
| ASTTE    | 152.10  | 262.13  | 15.79   | 137.46  | 244.47  | 14.80   |

**FIGURE 5.** (a) Prediction results of the trips which travel time around 20 minutes. (b) Prediction results of the trips which travel time around 40 minutes. (Hefei).

**FIGURE 6.** The MAPE(%) for trips with different time duration on the test set. (Hefei).

---

**FIGURE 7.** (a) Travel Time Prediction Results on Hefei dataset. (b) Travel Time Prediction Results on Chengdu dataset.
ASTTE-S which discards the speed information. Specifically, we replace the input of the LSTM in the Intersection Direction Field component from $\hat{r}_i$ with $r_i$ which only contains embedding of the underlying road network. A display of model performances through different times of the day shows in Fig. 8, overall, compared with model ASTTE that takes speed information into account, ASTTE-S performs poorly in one-day forecasts, besides, it also suffers more from morning and evening peaks.

**TABLE 4.** Performance of varying degree value of $K$. (Hefei).

| $K$   | MAE(%) | RMSE(%) | MAPE(%) |
|-------|--------|---------|---------|
| 0     | 295.01 | 16.73   |
| 1     | 277.89 | 16.28   |
| 2     | 270.92 | 15.79   |
| 3     | 262.86 | 15.92   |
| 4     | 271.52 | 16.42   |

5) **EFFECT OF GRAPH CONVOLUTION NETWORK**

Graph Convolution Network(GCN) is applied in the component of Road Segment Field to learn road network level information as a neighbor dependency. To study the effects of GCN in our proposed model ASTTE, we evaluate models by varying the filtering degree $K$. The results show in Table 4. ASTTE-G refers to the model directly uses road segment level information of yesterday to replace graph convolution operation to obtain the neighbor dependency, the other models represent the case with the value of $K$ gradually increases. It is obvious that the test metrics decrease with the help of graph convolution operation. Besides, We can see that when the value of $K$ near 2 or 3, our method achieves the best performance. When the value of $K$ is greater than 2, the performance slightly remains stable. The reason might lie that with the increase of $K$, the graph convolution operation tends to unify the whole road network traffic information which results in neglect of characteristics of the road segment involved in the trip.

6) **EFFECT OF INTERSECTION FIELD**

To study the effect of intersection direction information processed with convolutional layer of the proposed model and which is often ignored in other works. We evaluate the model by eliminating this component and utilize the information of road segments merely, named ASTTE-C. We test the impact of rating convergence with training epochs. Fig. 9 shows the result, as expected, both the model can achieve a low MAPE. However, it is obvious that the rate of declining on ASTTE-C tends to be slow and the integration of intersection information helps the model to make a more accurate prediction. The reason may be that the intersection information can better help the model identify whether it is a congested area so as to converge with a faster speed.

**TABLE 5.** Performance of ASTTE without attention mechanism.

| City    | MAE(%) | RMSE(%) | MAPE(%) |
|---------|--------|---------|---------|
| Hefei   | 173.09 | 279.98  | 16.35   |
| Chengdu | 152.56 | 268.14  | 16.04   |

7) **EFFECT OF ATTENTION MECHANISM**

It can be found by comparing the results in Table 3 and Table 5, with the attention mechanism, our model further reduces MAPE from 16.35 to 15.79 in the Hefei dataset and reduces MAPE from 16.04 to 14.80 in the Chengdu dataset, indicating the attention mechanism applied in ASTTE is efficient in capturing the dynamically changing spatial-temporal patterns in traffic data. Table 1 shows an example of a trip travels in the path of Hefei, which involves 11 road segments and 10 intersections. Fig. 10 presents the weights of each
attention channel of this trip, \( \hat{h}_1 \)-\( \hat{h}_{11} \) denote the output of the Road Segment Field component, \( \hat{h}_{12} \)-\( \hat{h}_{17} \) present the key features vectors obtained in the Intersection Direction Field component. It can be seen from the figure that the weight of the start and end road sections is greater, and the weight of the road section is greater than the weight of the intersection.

VI. CONCLUSION

In this paper, we make some works on the problem of route travel time estimation on taxi trajectory data, which provides a fundamental reference for intelligent transportation systems management. Specifically, we present a new perspective to represent the real-world road network as multi-relational data by taking real-world restrictions into consideration. We propose a new neural embedding method that generates road segments and intersection directions vectors of road network structure to model the spatial dependency which is closer to the real topology. In addition, for the particularity of time prediction, we learn the traffic information from the road segment and road network aspect and integrate the complex factors effectively making the model more robust. The attention mechanism allows the model to focus on some significant elements such as crowded road segments or intersections. For future work, we will try some better ways to represent the road network structure and may model the topology to one or many to many relationships. Besides, we will consider constructing a more complex structure to better integrate the information of road segments and intersections.

REFERENCES

[1] Y. Li, D. Deng, U. Demiryurek, C. Shahabi, S. Ravada, “Towards fast and accurate solutions to vehicle routing in a large-scale and dynamic environment,” in Advances in Spatial and Temporal Databases, C. Clarumun, M. Schneider, R. C.-W. Wong, L. Xiong, W.-L. Loh, C. Shahabi, and K.-J. Li, Eds, Cham, Switzerland: Springer, 2015, pp. 119–136.

[2] Z. Zou, H. Yang, and A.-X. Zhu, “Estimation of travel time based on ensemble method with multi-modality perspective urban big data,” IEEE Access, vol. 8, pp. 24819–24828, 2020.

[3] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[4] L. Li and W. Liu, “Travel time prediction on un-monitored roads: A spatial factorization machine based approach (student abstract),” in Proc. AAAI Conf. Intell. Intell., vol. 34, no. 10, 2020, pp. 13855–13856.

[5] Y. Duan, Y. L. V., and F.-Y. Wang, “Travel time prediction with LSTM neural network,” in Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2016, pp. 1053–1058.

[6] C.-H. Wu, J.-M. Ho, and D. T. Lee, “Travel-time prediction with support vector regression,” IEEE Trans. Intell. Transp. Syst., vol. 5, no. 4, pp. 276–281, Dec. 2004.

[7] Y. Wang, Y. Zheng, and Y. Xue, “Travel time estimation of a path using sparse trajectories,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining - KDD, 2014, pp. 25–34.

[8] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, “A simple baseline for travel time estimation using large-scale trip data,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, pp. 1–22, Feb. 2019.

[9] E. Jelencius and H. N. Koutsopoulos, “Travel time estimation for urban road networks using low frequency probe vehicle data,” Transp. Res. B, Methodol., vol. 53, pp. 64–81, Jul. 2013.

[10] R. Gao, X. Guo, F. Sun, L. Dai, J. Zhu, C. Hu, and H. Li, “Aggressive driving saves more time? Multi-task learning for customized travel time estimation,” in Proc. 28th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 1689–1696.

[11] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, “Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction,” Sensors, vol. 17, no. 4, p. 818, Apr. 2017.

[12] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, “Multi-task representation learning for travel time estimation,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Jul. 2018, pp. 1695–1704.

[13] H. Zhang, H. Wu, W. Sun, and B. Zheng, “DeepTravel: A neural network based travel time estimation model with auxiliary supervision,” 2018, arXiv:1802.02147. [Online]. Available: http://arxiv.org/abs/1802.02147

[14] I. Jindal, Tony, Qin, X. Chen, M. Nokleby, and J. Ye, “A unified neural network approach for estimating travel time and distance for a taxi trip,” 2017, arXiv:1710.04350. [Online]. Available: http://arxiv.org/abs/1710.04350

[15] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, “When will you arrive? Estimating travel time based on deep neural networks,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 1–8.

[16] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1188–1196.

[17] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 2787–2795.

[18] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” 2013, arXiv:1312.6203. [Online]. Available: http://arxiv.org/abs/1312.6203

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining - KDD, 2014, pp. 701–710.

[20] K. Liu, Y. Li, F. He, J. Xu, and Z. Ding, “Effective map-matching on the most simplified road network,” in Proc. 20th Int. Conf. Adv. Geographic Information System - SIGSPATIAL, 2012, pp. 600–612.

[21] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, “Traffic speed prediction and congestion source exploration: A deep learning method,” in Proc. IEEE 16th Int. Conf. Data Mining (ICDM), Dec. 2016, pp. 499–508.

[22] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” 2017, arXiv:1709.04875. [Online]. Available: http://arxiv.org/abs/1709.04875

[23] N. Wu, J. Wang, W. X. Zhao, and Y. Jin, “Learning to effectively estimate the travel time for fastest route recommendation,” in Proc. 20th ACM Int. Conf. Inf. Knowl. Manage., Nov. 2019, pp. 1923–1932.

[24] J. L. Elman, “Finding structure in time,” Cognit. Sci., vol. 14, no. 2, pp. 179–211, Mar. 1990.

[25] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[26] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, arXiv:1408.5882. [Online]. Available: http://arxiv.org/abs/1408.5882

[27] G. Sun, Y. Cong, Y. Zhang, G. Zhao, and Y. Fu, “Continual multiview task learning via deep matrix factorization,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jun. 2020, doi: 10.1109/TNNLS.2020.2977497.

[28] J. Dong, Y. Cong, G. Sun, B. Zhong, and X. Xu, “What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4023–4032.
[29] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 1019–1027.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 2017, pp. 5998–6008.

ZHONG WANG received the B.S. degree from Anhui Normal University, China, in 2018. She is currently pursuing the M.S. degree with the University of Science and Technology of China, China. Her research interests include network embedding and deep learning.

HAO FU received the B.S. degree from Northwestern Polytechnical University, China, in 2018. He is currently pursuing the master’s degree with the School of Computer Science and Technology, University of Science and Technology of China, China. His main research interests include machine learning, deep learning, and data mining.

GUIQUAN LIU received the B.Sc. and Ph.D. degrees from the University of Science and Technology of China, in 1996 and 1999, respectively. He is currently an Associate Professor with the School of Computer Science and Technology, University of Science and Technology of China. His main research interests include machine learning, data mining, social network analysis, and pattern recommendation. He has published over 60 papers in refereed conferences and journals.

XIANWEI MENG (Member, IEEE) received the M.Sc. degree from the Shenyang Institute of Automation, Chinese Academy of Sciences, in 2005. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China. His main research interests include data mining, machine learning, and satellite navigation application. He has 30 patents granted, including nine invention patents.