Trajectory-User Linking with Attentive Recurrent Network

Congcong Miao[1,2,3], Jilong Wang[2,3,*], Heng Yu[2,3], Weichen Zhang[2,3], Yinyao Qi[2,3]
[1]Department of Computer Science and Technology, Tsinghua University, China,
[2]Institute for Network Sciences and Cyberspace, Tsinghua University, China,
[3] Beijing National Research Center for Information Science and Technology (BNRist), China
{mcc15,h-yu17}@mails.tsinghua.edu.cn,wjl@tsinghua.edu.cn,zhangweichen@bupt.edu.cn

ABSTRACT

Trajectory-User Linking (TUL), which links trajectories to users, is a recently introduced trajectory mining task with a wide spectrum of applications, ranging from personalized recommendation systems and location-based social networks to potential criminals identification. Previous approaches on solving the TUL task generally rely on classic models or Recurrent Neural Network (RNN) models. However, their prediction accuracy is always not satisfactory due to the following reasons: 1) trajectory data is low-sampling and sparse; 2) user mobility patterns are high-order and multi-periodic; and 3) previous approaches fail to utilize existing abundant features. In this paper, we propose DeepTUL, which is composed of a feature representation layer and a recurrent network with attention mechanism, to solve TUL task. DeepTUL not only combines multiple features that govern user mobility to model high-order and complex mobility patterns, but also learns from labeled historical trajectory to capture multi-periodic nature of user mobility and alleviate the data sparsity problem. Extensive experiments show that DeepTUL yields significant improvements over the existing methods on two types of real-life mobility datasets (i.e., check-in dataset and WLAN dataset). Moreover, DeepTUL provides intuitive interpretation into the trajectory-user linking.

KEYWORDS

Deep learning; Human mobility mining; Social tagging

ACM Reference Format:
Congcong Miao, JilongWang, Heng Yu, Weichen Zhang, Yinyao Qi. 2020. Trajectory-User Linking with Attentive Recurrent Network. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020.* IFAAMAS, 9 pages.

1 INTRODUCTION

The advance of GPS and Wi-Fi enabled mobile devices and the enhancing ability of system in collecting information, more and more spatio-temporal trajectory data has been collected. The massive trajectory data provides us with unprecedented information to understand human mobility patterns and stimulates a number of trajectory mining tasks for various applications [26]. For example, STRNN [8] and Deepmove [3] model sequential transition patterns from historical trajectory data and predict where a user will go next for personalized recommendation. Gmove [22] and PCRN [31] model group-level mobility patterns to understand populace flow and then take actions accordingly to prevent traffic jams. BDAC [9] models human mobility from the individual and crowd level, and then applies mobility patterns for efficient network address management.

Trajectory-User Linking (TUL), which links trajectories to users who generate them, is a recently introduced trajectory mining task [4]. It is an important aspect to understand user mobility patterns for various location-based applications. For example, Wi-Fi controller and ride-sharing (Uber, Mobike) apps always collect a large amount of trajectory data – but the user identities are often eliminated for the sake of privacy. However, linking trajectories to their corresponding users seems desirable for personalized recommendation systems and location-based social networks. Also, users always share locations among several online social networks, such as Facebook, Twitter and Instagram. It is in a great need of linking users from cross-platform trajectory data which would help identity verification and privacy protection [29]. Furthermore, it may help in identifying the potential criminals or terrorists from the trajectory data (e.g., the transient phone signals and check-ins) [4].

As for its wide applications, the TUL task has attracted much effort [19, 28, 29]. The main process of solving TUL task is to model trajectories for mining sequential transition patterns, and then link them to users based on these patterns. Existing solutions on mining sequential transition patterns are always based on sequential statistical methods such as Markov models and Recurrent Neural Network (RNN) models. Markov models [6] fall short in constructing effective relationships among long-range dependencies of trajectory. RNN models show promising performance in TUL problem [4] and can extract hierarchical mobility patterns with semi-supervised learning [28]. Despite the inspiring results of current RNN models, there remain three challenges to be solved to realize high predictability in TUL: (1) *data sparsity*: the available trajectory data is conducted by user on voluntary basis, thus low-sampling in nature [16]. (2) *high-order and multi-periodic mobility patterns*: human mobility patterns vary among the changes of contextual scenarios and are multi-periodic [3]. (3) *integrating more diverse features*: previous approaches only utilize spatial feature [4, 28] and fail to utilize existing abundant features in trajectory data such as temporal feature.

To address the aforementioned challenges, we propose DeepTUL, an attentive recurrent network model, to solve the TUL task. In DeepTUL, we design a feature representation layer to utilize multiple features that govern user mobility and convert these features into dense representations. Another key component of DeepTUL is the RNN and attention layer which models sequential transition patterns and multi-periodic mobility regularities. In detail, the Bidirectional Long Short Term Memory (BiLSTM) is trained to capture high-order and complex sequential information, while the attention

* Corresponding author.
Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

module is trained to capture the multi-periodic nature of user mobility from labeled historical trajectories (i.e. we have known who the trajectories belong to in historical trajectories). DeepTUL alleviates the data sparsity problem by leveraging historical data and flexibly takes advantage of periodic mobility regularity to augment the RNN models to improve prediction accuracy. As a summary, our main contributions are as follows:

- We propose DeepTUL, an attentive recurrent network model to solve the TUL task. Our model combines multiple features and learns multi-periodic nature of user mobility from labeled historical trajectories to realize high predictability. To the best of our knowledge, DeepTUL is the first model that utilizes multi-periodic mobility patterns for accurate trajectory-user linking.
- We design two aggregation strategies in a historical attention module to generate representative vector of historical trajectories. Both strategies reveal the periodicity of user mobility by scoring generated vector with the current vector. Furthermore, we propose using scaled dot products in score function, which promotes the performance of trajectory-user linking prediction.
- We conduct comprehensive experiments on two types of real-life mobility datasets (i.e. WLAN dataset and check-in dataset) to evaluate the effectiveness of our model. The results demonstrate that our model yields significant improvements over the existing methods. Furthermore, compared with existing methods, DeepTUL provides intuitive interpretation into the trajectory-user linking.

The rest of this paper is organized as follows. Section 2 reviews the related work and Section 3 presents the notations and problem formulation. In Section 4, we introduce the architecture of DeepTUL in detail. After that, we apply DeepTUL on two types of real-life mobility datasets in Section 5. Finally, we conclude our paper in Section 6.

2 RELATED WORK

Trajectory data provides us with unprecedented information to understand the human mobility patterns [26] and trajectory data mining is becoming an increasingly important research theme that attracts much attention from academic researches. Due to its broad range of applications in personalized recommender systems [2, 8, 17], location-based social networks [29] and smart city planning [14, 24], researchers have come up with a number of mining tasks, such as trajectory classification [27], trajectory pattern mining [22] and trajectory outlier detection [30]. Recently, a new mining task named TUL, which links trajectories to users who generate them, was introduced by TULER [4]. It is an important aspect to understand user mobility patterns for various location-based applications and gradually becomes a hot topic.

The key step on solving the TUL task is to model trajectories for mining sequential transition patterns. Markov Chains (MC) based models are the commonly-used classic methods for modeling sequential transition patterns [6]. In these methods, an estimated transition matrix is built to predict the future action of users. However, these models have difficulty in constructing effective relationships among long term dependencies of trajectory, so that they are hard to capture high-order and multi-periodic mobility regularities. Recently, Recurrent Neural Network (RNN) based models not only have been successfully applied in word embedding for sentence modeling [11], but also show great power to capture long term dependencies in the sequence processing [25]. In the light of this, TULER [4] took advantage of RNN models to mine sequential transition patterns from trajectories, and then linked them to users. Similar to word embedding, it embedded each location with dense representation to capture semantic information and fed them into the RNN model to extract complex mobility patterns for TUL prediction. However, the "vanilla" RNN based models suffered from data sparsity problem and lacking of understanding multi-periodic nature of user mobility. Another work by [28] improved the prediction accuracy in TUL by understanding hierarchical semantics of human mobility and addressing data sparsity problem via incorporating unlabeled data. However, it failed to utilize existing abundant features and also did not consider multi-periodic mobility regularities.

Directly applying standard RNN models in trajectory mining is an intuitive way but not an efficient way to study human mobility patterns. Attention mechanism, firstly introduced in seq2seq machine translation [1], is not only able to strengthen RNN models in capturing long-term dependencies, but also presents interpretable prediction. Motivated by this, attention mechanism is widely adopted in trajectory mining tasks in recent years such as next point prediction [3], destination prediction [23] and traffic prediction [20]. The results in these works have shown the superiority of the attention mechanism in trajectory mining tasks. To the best of our knowledge, we are the first to propose using the attention mechanism in the TUL problem which not only alleviates the data sparsity problem by leveraging historical data, but also captures multi-periodic regularities of human mobility to improve prediction accuracy.

3 NOTATIONS AND PROBLEM FORMULATION

The trajectory data is collected with millions of entries in a region. Each entry is recorded with spatial and temporal context to present the essential factors for an event, i.e., where and when [26]. We partition a region (e.g. city or campus) in a suitable coordinate system (e.g., longitude+latitude or some landmarks) to get a number of representative location points, namely point of interest (POI). Let $\mathcal{P}$ denote a set of POIs and $\mathcal{U}$ denote a set of users.

Definition 1. *(Spatio-temporal Point) The spatio-temporal point q is a tuple of timestamp t and POI $p \in \mathcal{P}$, i.e., $q = (t, p)$.*

Definition 2. *(Trajectory) Given a set of spatio-temporal points q_i generated by user $u_i \in \mathcal{U}$ in a w-th time interval, the trajectory is represented as $S^w_{u_i} = q_{i1}q_{i2}\cdots q_{im}$ where q is listed by timestamp.*

The time interval can be a quarter day, one day or even one week. Following [4], the time interval in our work is set to 6 hours. A trajectory, generated in the w-th time interval $\overline{s^w} = \{q_1, q_2, \cdots, q_m\}$ for which we do not know the user who generated it, is called *unlinked*. The historical trajectories of the unlinked trajectory can be represented as a collection of spatio-temporal points generated in previous time intervals by all users, i.e., $\mathcal{H}^w = H^{w-n}\cdots H^{w-2}H^{w-1}$,

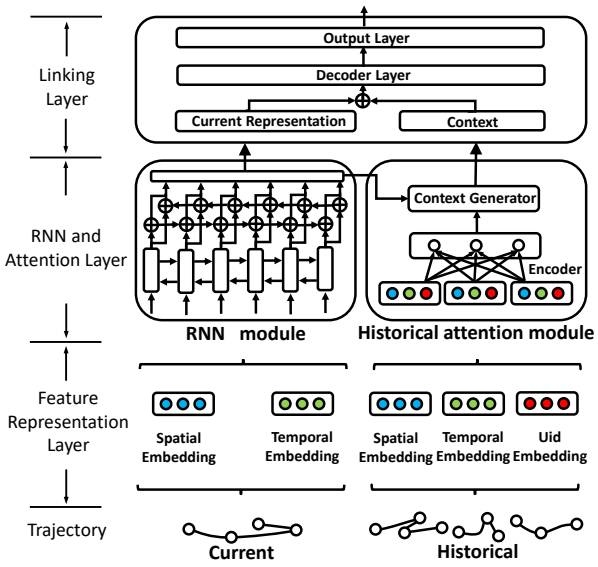


Figure 1: The architecture of DeepTUL.

where $H^w = \{S^w_{u_i} | u_i \in \mathcal{U}\}$ is the whole trajectories in the w-th time interval and n is the number of previous time interval.

Problem (TUL task): Given the current unlinked trajectory $\overline{s^w} = \{q_1, q_2, \cdots, q_m\}$ and corresponding historical trajectories $\mathcal{H}^w$, the TUL task aims to provide a mapping function that links the unlinked trajectory to users: $\overline{s^w} \rightarrow \mathcal{U}$.

4 DEEPTUL

In this section, we describe the details of our proposed DeepTUL. As shown in Figure 1, DeepTUL consists of three layers: *a feature representation layer*, *an RNN and attention layer* and *a linking layer*. The first layer aims at capturing multiple features that govern user mobility and converting these features into dense representations. Then, these represented features are fed into the middle layer to generate meaningful contexts of human mobility regularities, and the last layer links trajectories to users by combining these meaningful contexts.

4.1 Feature Representation Layer

The feature representation layer converts features that govern user mobility into dense representations. For the current trajectory, we learn embeddings for general features, i.e., location and timestamp, then concatenate them into a vector to generate a global representation. Note that for historical trajectories which are labeled by users, we learn an extra feature, i.e, uid embedding.

Location embedding. Each location point p is represented by a $|\mathcal{P}|$-dimensional one-hot vector. This phenomenon may result in the curse of dimensionality if the number of location points grows large. Motivated by the word embeddings in natural language processing [10], we represent each location with a low-dimensional vector with a transformation matrix E_p to capture precise semantic information. Specifically, for a raw input location point p represented by a one-hot vector, we transform p into a low-dimensional vector with a transformation matrix $E_p \in \mathbb{R}^{|\mathcal{P}| \times D_p}$, where D_p is the dimension for location embedding, i.e., $v_p = p \cdot E_p$. In location embedding, we try to learn the transformation matrix E_p.

Timestamp embedding. The original temporal information in the entry is a real-value timestamp. It is impracticable to directly embed each timestamp due to its continuous property. Therefore, we align all the timestamp into a fixed time interval. The time interval can be a day, two days (i.e. weekday and weekend) or a week. We then discretize the fixed time interval into T timeslots (a basic embedding unit) to aggregate the user mobility statistics. For a raw input timestamp t, we transform t into a one-hot T dimensional vector. The timestamp embedding tries to learn a transformation matrix $E_t \in \mathbb{R}^{T \times D_t}$, where D_t is the dimension for timestamp embedding. With the matrix E_t, we can transform timestamp t into a D_t-dimensional vector v_t based on the equation: $v_t = t \cdot E_t$.

Uid embedding. The historical trajectories are the labeled data which contains valuable user identification (label). With the help of user identification and spatial and temporal information in historical trajectories, we will be able to extract periodicity of user mobility. Similar to spatial and temporal embeddings, we try to learn a transition matrix $E_u \in \mathbb{R}^{|\mathcal{U}| \times D_u}$ to represent the users where $|\mathcal{U}|$ is the number of users and D_u is the dimensions for embedding. With the matrix E_u, we can transform one-hot vector u into a low-dimensional vector v_u based on the equation: $v_u = u \cdot E_u$.

We then concatenate the location embedding and timestamp embedding to generate a global representation $v_q \in \mathbb{R}^{D_q}$ where $D_q = D_p + D_t$. Previous works on solving TUL task only focus on spatial feature and the embedding vector of the spatial point was trained offline [4, 28]. In our work, we design a feature representation layer by jointly considering multiple features that govern user mobility.

4.2 RNN and Attention Layer

The RNN and attention layer consists of two modules: an RNN module and a historical attention module. The former uses Bidirectional Long Short Term Memory (BiLSTM) to capture high-order and complex sequential information in current trajectory, while the latter learns multi-periodic nature of user mobility from labeled historical trajectories to augment the RNN module for trajectory-user linking.

RNN module. Long Short Term Memory (LSTM) is a variant of RNN which can handle variable-length trajectories and avoid the gradient vanishing problem through the gating mechanism, and thus being more suitable for our problem. The architecture of LSTM consists of a memory cell and three gates with the flow of information inside the LSTM unit: an input gate i, an output gate o and a forget gate f at t-th input $x_t \in \mathbb{R}^{D_q}$:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1} + b_o) \\ f_t &= \sigma(W_t x_t + U_t h_{t-1} + V_f c_{t-1} + b_f) \end{aligned} \tag{1}$$

where σ is the sigmoid activation function; c_t represents the current cell state which is a combination of gates obtained in the following way

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2}$$

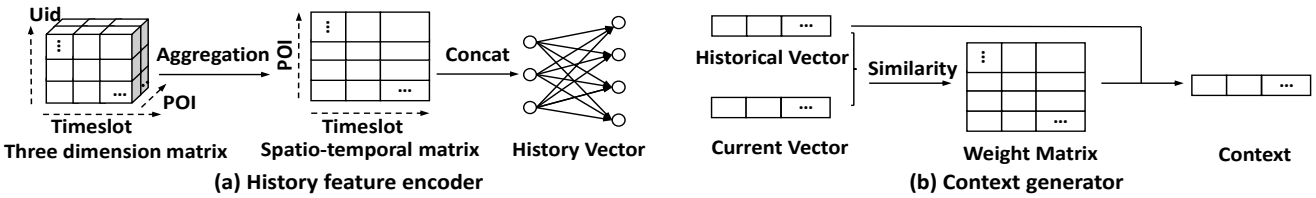


Figure 2: Historical attention module.

where $\odot$ is the element-wise product; $tanh(.)$ is the hyperbolic tangent function; the candidate state $h_t \in \mathbb{R}^{D_h}$ is generated by

$$\overrightarrow{h_t} = o_t \odot tanh(c_t) \tag{3}$$

where W, U and b are all parameterized matrices.

The general LSTM network only captures preceding part of the trajectory and ignores the future context in the trajectory. In order to enhance the memory capability, we use bidirectional LSTM (BiLSTM) which utilizes additional backward layers [5] to exploit the contexts from both the past and the future. In the t-th time, we get the $2D_h$-dimensional representation:

$$h_t = [\overrightarrow{h_t} \oplus \overleftarrow{h_t}] \tag{4}$$

Historical attention module. Despite the good learning of BiLSTM in the RNN module in modeling sequential transition patterns, it is unable to capture periodicity of user mobility. The attention mechanism is introduced to augment RNN models and has recently demonstrated success in trajectory mining tasks [3, 23]. The attention mechanism in our work takes the historical trajectories as the input vector, takes the current trajectory as queried vector and outputs the most related context vector as the periodicity representation. As is shown in Figure. 2, the historical attention module consists of two main parts: historical feature encoder and context generator. The former is to generate the historical vector which represents the mobility regularities of users as the candidate vectors, and the latter is to identify the vector that is closely related to the queried vector.

1) Historical feature encoder The main idea of historical feature encoder is to represent historical trajectories into historical vectors. A basic way is to use an RNN based method which takes the historical trajectory of each user as input and outputs the representation of each user. Then a concatenation operation, which concatenates the representations by users, is used to generate historical vectors. However, it does not directly simulate the periodicity of human mobility[3]. To solve this problem, we design a historical feature encoder method which is shown in Figure 2(a). We firstly reorganize historical trajectories into three dimensions with spatial dimension, temporal dimension and user dimension. Then we align all the timestamp into a fixed time interval. We collect the users fall into the same spatio-temporal dimension to generate a visited user list L. For example, if users u_1 and u_2 visit location p_1 at timeslot t_1, the visited user list $L_{p1,t1}$ of that spatio-temporal point can be represented as $[u_1, u_2]$. If user u_1 visits location p_1 at timeslot t_1 and users u_1 and u_2 visit the same location at timeslot t_1 a week later, the three records fall into the same spatio-temporal dimension and the visited user list $L_{p1,t1}$ can be represented as $[u_1, u_1, u_2]$. The spatio-temporal dimension is used to capture the spatio-temporal preference and simulate the periodicity of user mobility.

In order to generate a representative spatio-temporal matrix that contains meaningful visited user information, we aggregate the users falling into the same spatio-temporal point. We design two aggregation strategies: (1) average aggregation strategy (AVE) and (2) maximum aggregation strategy (MAX); The average aggregation strategy adds up all the vectors which is generated by the uid embedding and then calculates their mean values as the representative vector. The maximum aggregation strategy extracts the most frequent user appeared in the list and then embeds the user as the representative vector. The representative vector can be computed as:

$$v^{\overline{u}}_{p_i,t_j} = \begin{cases} \sum\limits_{u_k \in L_{p_i,t_j}} u_k \cdot E_u / \left|L_{p_i,t_j}\right|, & \text{AVE} \\ max(L_{p_i,t_j}) \cdot E_u, & \text{MAX} \end{cases} \tag{5}$$

where $v^{\overline{u}}_{p_i,t_j} \in \mathbb{R}^{D_u}$ is the representative vector of spatio-temporal point (p_i, t_j); E_u is a user transformation matrix and $max(L_{p_i,t_j})$ is the most frequent user appeared in the list L_{p_i,t_j}. The performance of using different aggregation strategies will be investigated and discussed in the experimental results and analysis section.

After that, we concatenate the spatio-temporal representative vector with location embedding and time embedding to generate a historical global representation $v_{u,p_i,t_j} \in \mathbb{R}^D$ where $D = D_u + D_p + D_t$. Finally, a fully connection layer is used to further process the historical global representation into the appropriate shape:

$$h_{i,j} = \sigma(W_h \cdot v_{u,p_i,t_j} + b_h) \tag{6}$$

where $h_{i,j} \in \mathbb{R}^{2D_h}$ is reshaped vector; $W_h \in \mathbb{R}^{2D_h \times D}$ and $b_h \in \mathbb{R}^{2D_h}$ are weight matrix and bias and activate function σ is the hyperbolic tangent function. Note, the dimension of $h_{i,j}$ should be the same as the dimension h_t generated in RNN module.

2) Context generator As shown in Figure 2(b), the context generator uses both current vector and historical vector to calculate similarity and form an attention weight matrix. Then, it generates the context vector which is formed by a weighted sum. The formulations are as follows:

$$a_{i,j} = \frac{exp(f(h_t, h_{i,j}))}{\sum_{i1}\sum_{j1} exp(f(h_t, h_{i1,j1}))} \tag{7}$$

$$c = \sum_i \sum_j a_{i,j} h_{i,j} \tag{8}$$

where h_t is the queried vector generated in RNN module; $h_{i,j}$ is the historical global representation vector of spatio-temporal point (p_i, t_j) generated in historical feature encoder; $a_{i,j}$ is the attention weight of spatio-temporal point (p_i, t_j) and c is the context vector which is formed by a weighted sum of historical data. The

Algorithm 1: Training algorithm for DeepTUL

Input: Time slice $\mathcal{W}$; Unlinked Trajectory set $\overline{S}$; Historical Trajectory Set $\mathcal{H}$
Output: Trained Model $\mathcal{M}$

// construct training instances;
$\mathcal{D} \leftarrow \emptyset$;
for $w \in \{1, 2, \cdots, \mathcal{W}\}$ **do**
  $\mathcal{H}^w = H^{w-n} \cdots H^{w-2} H^{w-1}$;
  for $k \in \{1, 2, \cdots, K\}$ **do**
    //K is the number of trajectory in w-th time interval;
    $cur = \overline{S_k^w}$; $his = \mathcal{H}^w$;
    put a training instance (w, cur, his) into $\mathcal{D}$;
initialize the parameter Θ;
for $i \in \{1, 2, \cdots, EPOCH\}$ **do**
  randomly select a batch of instances $\mathcal{D}_b$ in order of time w from $\mathcal{D}$;
  update Θ by minimizing the objective (12) with $\mathcal{D}_b$;
  stop training when criteria is met;
output trained model $\mathcal{M}$;

score function $f(.)$ is dot products to calculate similarity of two vectors. However, for large dimension of vectors, the dot products grow large in magnitude, pushing Eq. 7 into regions where it has extremely small gradients [15]. To counteract this effect, we scale the dot products by $\frac{1}{\sqrt{2D_h}}$. The formulation is as follows:

$$f(h_t, h_{i,j}) = \frac{h_t h_{i,j}^T}{\sqrt{2D_h}} \tag{9}$$

where $2D_h$ is the dimension of vectors. We will evaluate the effectiveness of scaled dot products in the experimental results and analysis section.

4.3 Linking layer

The linking layer aims to get the probability distribution of users (labels) from the high-dimension context generated in RNN and historical attention layer. The linking layer firstly concatenates current representation h_t from RNN module and the context c from historical attention module to get more high-level representations. Then, the vector is fed into the decoder layer to project high-level representations into a vector with $\mathcal{U}$ dimension. The formulation is as follows:

$$\mathbf{y} = (W_c \begin{bmatrix} h_t \\ c \end{bmatrix} + b_c) \tag{10}$$

where $W_c \in \mathbb{R}^{|\mathcal{U}| \times 4D_h}$ and $b_c \in \mathbb{R}^{|\mathcal{U}|}$ are weight matrix and bias; The vector $\mathbf{y}$ in decoder layer is associated with the probability of each user where $\mathbf{y}_i$ represents the probability of user u_i.

4.4 Network Training

To train our proposed DeepTUL model, we transform the values in output layer to normalized probabilities by:

$$p(y \mid \overline{s^w}, \mathcal{H}^w) = \sigma(\mathbf{y}) \tag{11}$$

where $\sigma(x)$ is the softmax function. The predicted label can be generated with $\hat{y} = argmax_y p(y \mid \overline{s^w}, \mathcal{H}^w)$. Generally, human mobility prediction tasks can be regarded as a multi-classification problem, so we apply cross-entropy as our loss function and use backward-propagation-through-time algorithm [18] and Adam [7] to train our model. The cost function is the negative log-likelihood of the true user label:

$$\mathcal{L}(\Theta) = -\frac{1}{l} \sum_{i=1}^{l} d_i log(\sigma(y_i)) + \frac{\lambda}{2} \|\Theta\|^2 \tag{12}$$

where $d_i \in \mathbb{R}^{|\mathcal{U}|}$ is the one-hot represented ground truth of the trajectory and $y_i \in \mathbb{R}^{|\mathcal{U}|}$ is the estimated probability of users; $\Theta = \{W, U, b, E_u, E_t, E_p\}$ are trainable parameters; l is the number of training trajectories; λ is an L2 regularization hyperparameter to alleviate overfitting problem. Algorithm 1 outlines the training process of DeepTUL.

Table 1: The description and statistics of two datasets. $|\mathcal{U}|$: number of users; $|\mathcal{T}_t|/|\mathcal{T}_e|$: number of trajectories for training and testing; $|\mathcal{P}|$: number of trajectory point; $|\mathcal{R}|$: average length of trajectories (before segmentation); $|\mathcal{T}_r|$: range of the trajectory length.

Dataset	$\|\mathcal{U}\|$	$\|\mathcal{T}_t\|/\|\mathcal{T}_e\|$	$\|\mathcal{P}\|$	$\|\mathcal{R}\|$	$\|\mathcal{T}_r\|$
Check-in	209	16151/4144	2877	191	[1,27]
	108	8681/2226	2166	204	[1,17]
WLAN	209	9632/2503	33	387	[1,42]
	108	4976/1294	33	411	[1,38]

5 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we conduct extensive experiments to demonstrate the effectiveness of DeepTUL on solving TUL task. We first introduce the datasets, baseline models and evaluation metrics of our experiments. Then we compare our DeepTUL with the baseline models and present analysis of impact of the different strategies. Finally, we present intuitive interpretation of the prediction.

5.1 Experimental Settings

Datasets: Two types of real-life mobility datasets are used in our experiment.

- **Check-in dataset** records user check-in behavior on laction-based social networks, containing detailed timestamps and coordinates. There are several check-in datasets publicly available online. The Foursquare [1] dataset, which has been used in previous works [28], is chosen in our work.
- **WLAN dataset** is the one-moth accessing records collected from a large scale WLAN in campus. The access controller (AC), controlling WiFi connection for a specific area, records user information with timestamp when users open WiFi for wireless communication. We regards each AC as POI in our work. The sensitive information is anonymized before it is used in our work.

Similar to [28], we randomly select $|\mathcal{U}|$ users and their corresponding trajectories from two dataset for evaluation. In order to check the robustness of our model, we select two different number of users from both datasets. On both datasets, we use the first 80% of each users' trajectory as training data. The remaining 20% of

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

Table 2: Performance comparison on two types of real-life mobility datasets evaluated by ACC@K, macro-P, macro-R, macro-F1

Dataset	Method	ACC@1	ACC@5	macro-P	macro-R	macro-F1	ACC@1	ACC@5	macro-P	macro-R	macro-F1
		$\|\mathcal{U}\|$=108					$\|\mathcal{U}\|$=209				
WLAN	TULER-LSTM	23.20%	52.75%	12.70%	16.58%	14.38%	17.63%	41.05%	8.60%	11.93%	9.99%
	TULER-GRU	24.21%	52.82%	15.45%	17.10%	16.23%	17.71%	41.41%	9.76%	11.93%	10.74%
	BiTULER	26.14%	56.46%	16.29%	18.89%	17.49%	18.86%	43.65%	11.34%	13.22%	12.21%
	TULVAE	26.68%	56.15%	16.50%	19.26%	17.77%	19.38%	44.32%	12.60%	13.80%	13.17%
	Attn-LSTM	36.53%	**70.12%**	27.91%	27.60%	27.75%	33.25%	62.27%	**25.59%**	25.53%	**25.56%**
	Attn-GRU	35.76%	68.89%	28.77%	28.71%	28.74%	33.33%	60.51%	24.07%	25.50%	24.76%
	DeepTUL	**38.54%**	68.27%	**29.46%**	**30.18%**	**29.82%**	**34.69%**	**65.47%**	23.80%	**26.32%**	25.00%
Check-in	TULER-LSTM	63.10%	74.11%	58.62%	51.77%	54.98%	58.32%	71.18%	51.30%	45.57%	48.27%
	TULER-GRU	63.33%	74.88%	**60.64%**	52.32%	56.17%	58.00%	71.69%	**53.33%**	45.46%	49.09%
	BiTULER	63.37%	74.92%	58.11%	52.44%	55.13%	58.75%	72.36%	52.13%	47.08%	49.47%
	TULVAE	63.64%	74.79%	58.49%	52.68%	55.43%	59.04%	72.46%	52.18%	47.42%	49.69%
	Attn-LSTM	68.12%	80.30%	56.85%	52.64%	54.66%	62.75%	74.77%	52.42%	47.45%	49.81%
	Attn-GRU	66.70%	**81.02%**	55.49%	51.45%	53.39%	63.06%	75.20%	52.22%	47.65%	49.83%
	DeepTUL	**69.24%**	80.61%	58.56%	**54.50%**	**56.46%**	**63.06%**	**75.57%**	52.66%	**48.03%**	**50.24%**

each users' trajectory are used as testing data for evaluating the model's performance. Table 1 reports the detailed descriptions and statistics of two datasets.

Hyperparameters: Our DeepTUL model is implemented on NVIDIA TITAN XP GPU using PyTorch and trained with Adam algorithm [7]. The location embedding D_p, timestamp embedding D_t and uid embedding D_u is set to 64, 32 and 32 respectively and uniformly initialized. The hidden size D_h is set to 64. The learning rate is set to 0.005 with the decays of 0.5. The number of previous time interval m is set to 28 for WLAN dataset and 120 for check-in dataset respectively. To avoid overfitting problem, we use two regularization techniques: the L2 regularization is set to 0.0002 for all trainable parameters and dropout technique is set to 0.6 in fully connected layer.

Metrics: The TUL task can be regarded as a multi-classificaition problem. We use ACC@K, macro-P, macro-R and macro-F1 to evaluate the performance, which are common metrics in multi-classificaition area . Specifically, ACC@K is to evaluate the accuracy of prediction. It is considered correct if the ground truth user $u^*(s)$ lies within the top-k user set $U_K(s)$. The formulation can be represented as:

$$ACC@K = \frac{|\{s \in S : u^*(s) \in U_K(s)\}|}{|S|} \tag{13}$$

where $|S|$ is the evaluation corpus. ACC@1 is the straightforward performance indicator which is the ratio of correct predictions. Macro-F1 is the harmonic mean of macro-P and macro-R which are averaged across all classes. It can be regarded as an overall performance indicator. The formulation is as follows:

$$\text{macro-F1} = \frac{2 \times \text{macro-P} \times \text{macro-R}}{\text{macro-P} + \text{macro-R}} \tag{14}$$

Approach for comparison: We compare DeepTUL with classic models as well as recent deep learning based models for trajectory-user linking. Specifically, the following baseline approaches are evaluated.

Classic models:

- **LCSS**: The Longest Common Sub-Sequence [21] is a widely used method to find the similarity among two sequences. We apply LCSS to TUL problem by searching the most similar trajectory in training set to find the corresponding user.
- **LDA**: The Linear Discriminant Analysis is a popular classification method used in spatial data [12]. We apply LDA to TUL problem by embedding trajectory into one-hot vectors and using SVD to decompose the within-class scatter matrix.
- **SVM**: The Support Vector Machine is a commonly used model in trajectory mining tasks [13]. We use the linear kernel in TUL problem which shows better performance than RBF and Gaussian kernel.

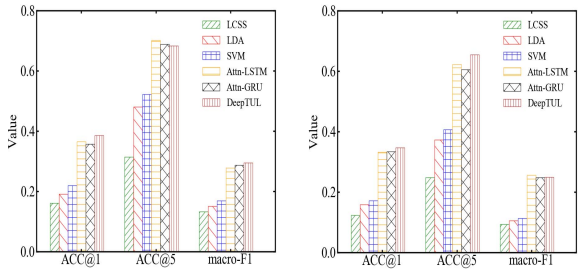
Deep neural network models:

- **RNN**: RNN, proposed in TULER [4], is the original method to solve TUL problem. There are several variants in TULER, such as RNN with Gated Recurrent Unit (**TULER-GRU**), RNN with Long Short-Term Memory (**TULER-LSTM**) and RNN with bidirectional LSTM (**BiTULER**).
- **TUVAE**: RNN with Variational AutoEncoder (VAE) is a state-of-art method [28]. It utilizes the ability of VAE to learn the hierarchical semantics of trajectory with stochastic latent variables that span hidden states in RNN.

For our models, we implement the variation in RNN module with gated recurrent unit and long short term memory unit, defined as (***Attn-LSTM***) and (***Attn-GRU***) respectively. Note, ***DeepTUL*** implements the Bidirectional LSTM in RNN module to enhance the memory capability.

5.2 Analysis of Experimental Results

The performance comparisons on two types of real-life datasets evaluated by ACC@K, macro-P, macro-R and macro-F1 are presented in Table 2, where the best value is shown in **bold** and the second best is shown as underlined. Since the comparison results have already shown the superior of RNN based models on Check-in dataset [4], we demonstrate the comparison with classic models in Figure 3 only for WLAN dataset. We observe that DeepTUL



(a) $|\mathcal{U}|$=108 (b) $|\mathcal{U}|$=209

Figure 3: Performance comparison with classic models on WLAN.

with BiLSTM and historical attention module shows superiority in capturing high-order and complex mobility patterns and presents performance improvements over classic models. As for comparing with RNN based models in Table 2, we observe that attentive recurrent networks (i.e. LSTM-attn, GRU-attn and DeepTUL) perform the best on most of metrics. This superior result is due to its capability of learning from historical trajectories. By modeling historical trajectories, our attention module learns multi-periodic nature of user mobility on each spatio-temporal point, and then generates the most related context of current trajectory to improve the prediction accuracy. On the other hand, DeepTUL alleviates the data sparsity problem by utilizing labeled historical data. Furthermore, we observe that the DeepTUL outperforms the LSTM-attn and GRU-attn. This phenomenon is due to the strong memory capability with BiLSTM which improves the prediction accuracy.

5.3 Impact of different strategies

As aforementioned, we can use different aggregation strategies (i.e., MAX or AVE) and score functions (i.e., dot or scaled dot) in the historical attention module to get the context. We focus on two representative metrics, straightforward performance indicator ACC@1 and overall performance indicator macro-F1, to conduct the evaluation. The result of using different methods among WLAN and check-in datasets are shown in Table 3. For example, *MAX-dot* denotes DeepTUL with maximum aggregation strategy and dot products enabled and *AVE-sdot* denotes DeepTUL with average aggregation strategy and scaled dot products enabled. For both datasets, we observe that *AVE* based DeepTUL outperforms *MAX* based DeepTUL. This can be explained by the reason that the maximum aggregation strategy loses users who rarely appear while the average aggregation strategy could retain more user information. Another observation is that *sdot* based DeepTUL outperforms *dot*

Table 3: Acc@1 and macro-F1 vs. different strategies

Strategy	WLAN($\|\mathcal{U}\|$=209)		Check-in($\|\mathcal{U}\|$=108)	
	Acc@1	macro-F1	Acc@1	macro-F1
MAX-dot	0.2750	0.1824	0.6690	0.5398
MAX-sdot	0.3213	0.2361	0.6792	0.5530
AVE-dot	0.2758	0.1837	0.6701	0.5489
AVE-sdot	0.3469	0.2500	0.6924	0.5646

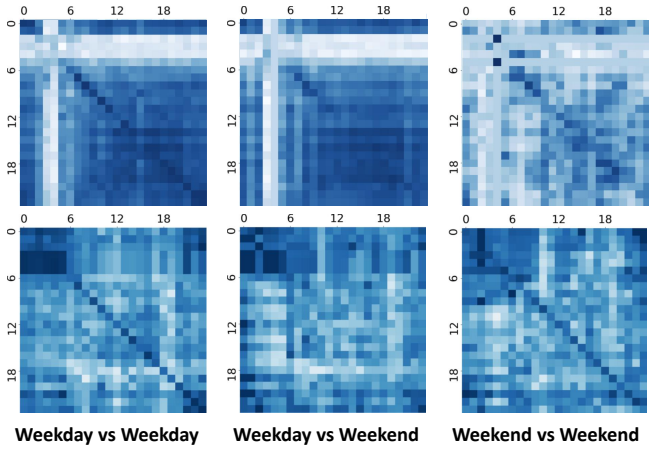


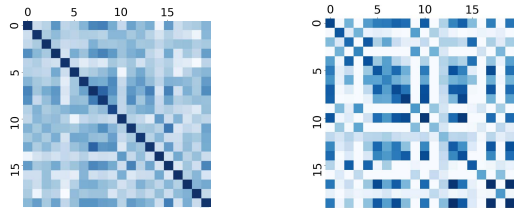
Figure 4: Heatmap comparison on Check-in (top) dataset and WLAN (botton) dataset.

based DeepTUL. This phenomenon verifies the importance of scaled dot products to counteract the effect of extremely small gradients. Among the different strategies, the *AVE-sdot* based DeepTUL is best, both in terms of Acc@1 and macro-F1.

5.4 Interpretation

Interpretation of attention module: We note the improvements due to the attention module when focusing more specifically on the comparison between TULER and attentive recurrent networks (i.e. attn-LSTM, attn-GRU and DeepTUL) in Table 2. In order to interpret whether user mobility exhibits periodic patterns, we split trajectories of each user into two parts: current trajectories and historical trajectories, and align them on the temporal dimension (i.e., [0,24]). For each user, we firstly calculate the weighted sum of location points falling in the same timeslot based on location transformation matrix generated after training of the model, and then calculate cosine similarity of current trajectories and historical trajectories. Figure 4 draws the heatmap. The horizon axis and vertical axis of each square matrix are both the time of a day. The shed of grid describes the similarity, where the deeper blue means higher similarity. For example, in the top-left square matrix, the entries in the diagonal are larger than other entries on most of cases, which indicates the user mobility shows daily regularity in weekdays. The top-right square matrix shows the similar result for weekends. However, the top-middle square matrix shows that user mobility patterns are different in weekday vs weekend which indicates user mobility regularities are multi-periodic. The bottom figures show the similar results in WLAN dataset.

Interpretation of prediction values: Another observation is that the metric values in WLAN dataset are much smaller than that in Check-in dataset in Table 2. In order to present interpretation, we randomly select 20 users in both datasets and split their trajectories into two equal parts to calculate the trajectory similarity of each user pair. Figure 5 plots cosine similarity of user mobility where horizon axis and vertical axis both represent user ID. The shed of grid describes the similarity where the deeper blue means higher similarity. We observe that the diagonal entries are larger than other



(a) Check-in (b) WLAN

Figure 5: Heatmap comparison on the similarity of user mobility.

entries on most of cases, which indicates that user mobility pattern follows a certain regularity. However, the diagonal entries in WLAN dataset seem not remarkable compared to their horizontal and vertical values. This phenomenon indicates more similar mobility patterns of users in WLAN. Therefore, the metric values decrease in WLAN dataset. For example, DeepTUL fails to link the trajectory of user ID 17 correctly, mainly because of the extremely similar mobility pattern of user ID 13 and user ID 19.

6 CONCLUSION

This paper has studied a popular task on trajectory data mining, namely trajectory-user linking (TUL). We propose an attentive recurrent network model DeepTUL, which is composed of a feature representation layer to capture multiple features that govern user mobility and a recurrent network with attention mechanism to capture multi-periodic nature of user mobility. It includes a feature representation layer to capture multiple features and converts them into dense representations and then utilizes an attention mechanism to augment the RNN model to improve performance. Two aggregation strategies are used in historical attention module to generate the representative vector of history trajectory. We conduct extensive experiments on two real-life mobility datasets to demonstrate the effectiveness of our model and present intuitive interpretation. In the future, it would be interesting to investigate more advanced models with more features to improve the performance of TUL.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Qiang Gao from University of Electronic Science and Technology of China for providing the python code of the *LDA* and *SVM* methods. The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. This work is supported by the National Key R&D Program of China (2016YFB0801301) and (2016QY12Z2103).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Chaochao Chen, Ziqi Liu, Peilin Zhao, Jun Zhou, and Xiaolong Li. 2018. Privacy Preserving Point-of-Interest Recommendation Using Decentralized Matrix Factorization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[3] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1459–1468.

[4] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying human mobility via trajectory embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 1689–1695.

[5] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610.

[6] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 191–200.

[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[8] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[9] Congcong Miao, Jilong Wang, Tianying Ji, Hui Wang, Chao Xu, Fenghua Li, and Fengyuan Ren. 2019. BDAC: A Behavior-aware Dynamic Adaptive Configuration on DHCP in Wireless LANs. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 1–11.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[11] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

[12] Hamid Reza Shahdoosti and Fardin Mirzapour. 2017. Spectral–spatial feature extraction using orthogonal linear discriminant analysis for classification of hyperspectral data. *European Journal of Remote Sensing* 50, 1 (2017), 111–124.

[13] Antonio F Skarmeta-Gomez. 2018. Classification of Spatio-Temporal Trajectories Based on Support Vector Machines. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection: 16th International Conference, PAAMS 2018, Toledo, Spain, June 20–22, 2018, Proceedings*, Vol. 10978. Springer, 140.

[14] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level.. In *IJCAI*, Vol. 16. 2618–2624.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[16] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. 2015. Geo-SAGE: A geographical sparse additive generative model for spatial item recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1255–1264.

[17] Weiqing Wang, Hongzhi Yin, Shazia Sadiq, Ling Chen, Min Xie, and Xiaofang Zhou. 2016. Spore: A sequential personalized spatial item recommender system. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 954–965.

[18] Paul J Werbos et al. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

[19] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. 2017. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1241–1250.

[20] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling Spatial-Temporal Dynamics for Traffic Prediction. *arXiv preprint arXiv:1803.01254* (2018).

[21] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S Tseng. 2010. Mining user similarity from semantic trajectories. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*. ACM, 19–26.

[22] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. 2016. Gmove: Group-level mobility modeling using geo-tagged social media. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1305–1314.

[23] Hanyuan Zhang, Hao Wu, Weiwei Sun, and Baihua Zheng. 2018. DeepTravel: a Neural Network Based Travel Time Estimation Model with Auxiliary Supervision. *arXiv preprint arXiv:1802.02147* (2018).

[24] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, and Tianrui Li. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence* 259 (2018), 147–166.

[25] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[26] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 29.

[27] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008. Understanding mobility based on GPS data. In *Proceedings of the 10th international*

conference on Ubiquitous computing. ACM, 312–321.

[28] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2018. Trajectory-User Linking via Variational AutoEncoder.. In *IJCAI*. 3212–3218.

[29] Fan Zhou, Lei Liu, Kunpeng Zhang, Goce Trajcevski, Jin Wu, and Ting Zhong. 2018. DeepLink: A Deep Learning Approach for User Identity Linkage. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1313–1321.

[30] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. 2017. Effective and efficient trajectory outlier detection based on time-dependent popular route. *World Wide Web* 20, 1 (2017), 111–134.

[31] Ali Zonoozi, Jung-jae Kim, Xiao-Li Li, and Gao Cong. 2018. Periodic-CRN: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns.. In *IJCAI*. 3732–3738.