DPLink: User Identity Linkage via Deep Neural Network From Heterogeneous Mobility Data

Jie Feng[1], Mingyang Zhang[1], Huandong Wang[1], Zeyu Yang[1], Chao Zhang[2],
Yong Li[1], Depeng Jin[1]
[1]Beijing National Research Center for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
liyong07@tsinghua.edu.cn

ABSTRACT

Online services are playing critical roles in almost all aspects of users' life. Users usually have multiple online identities (IDs) in different online services. In order to fuse the separated user data in multiple services for better business intelligence, it is critical for service providers to link online IDs belonging to the same user. On the other hand, the popularity of mobile networks and GPS-equipped smart devices have provided a generic way to link IDs, *i.e.*, utilizing the *mobility traces* of IDs. However, linking IDs based on their mobility traces has been a challenging problem due to the highly heterogeneous, incomplete and noisy mobility data across services.

In this paper, we propose *DPLink*, an end-to-end deep learning based framework, to complete the user identity linkage task for heterogeneous mobility data collected from different services with different properties. *DPLink* is made up by a *feature extractor* including a location encoder and a trajectory encoder to extract representative features from trajectory and a *comparator* to compare and decide whether to link two trajectories as the same user. Particularly, we propose a pre-training strategy with a simple task to train the *DPLink* model to overcome the training difficulties introduced by the highly heterogeneous nature of different source mobility data. Besides, we introduce a multi-modal embedding network and a co-attention mechanism in *DPLink* to deal with the low-quality problem of mobility data. By conducting extensive experiments on two real-life ground-truth mobility datasets with eight baselines, we demonstrate that *DPLink* outperforms the state-of-the-art solutions by more than 15% in terms of hit-precision. Moreover, it is expandable to add external geographical context data and works stably with heterogeneous noisy mobility traces. Our code is publicly available[1].

CCS CONCEPTS

• **Information systems → Location based services**; **Data mining**; • **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**;

KEYWORDS

deep learning; mobility trajectory; user identity linkage

ACM Reference Format:
Jie Feng[1], Mingyang Zhang[1], Huandong Wang[1], Zeyu Yang[1], Chao Zhang[2], Yong Li[1], Depeng Jin[1] . 2019. *DPLink*: User Identity Linkage via Deep Neural Network From Heterogeneous Mobility Data. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313424

1 INTRODUCTION

Smartphones and other mobile devices have made it easy for users to access various online services nearly everywhere and at any time. It is very common for a user to have multiple online identifiers (IDs) in different services such as online social networks (OSN), e-commerce services, online games, etc. Service providers have strong motivations to massively mine user data for monetization and optimizing user experience [16]. To capture a more comprehensive understanding of user behavior, it is increasingly intriguing to link user IDs across multiple services to fuse the separated data [37, 40].

To these ends, linking online IDs plays a critical role in the data fusion for better business intelligence. Early research has explored different ways to link user IDs by using service-specific data such as user profile attributes [9] and social graphs [15]. However, these approaches depend on whether these services have the same data type. For example, e-commerce services often do not have social graphs to match with an online social network. Moreover, users may fill in the fake information (*e.g.*, name, gender) in their profiles, which makes the linkage even harder.

In this paper, we explore a more generic approach to link user IDs by leveraging the spatial-temporal locality of user activities. The key intuition is that no matter what online services a user accesses, we can bind them to the user's *physical presence*, which is characterized by time and location. This becomes possible because most online services today have a mobile version with the location as parts of the service (*e.g.*, Uber, Yelp, Twitter). In addition, with some tolerance on granularity, even network accessing related information can be translated into location [14]. Our goal is to link multiple online IDs that belong to the same users across different services. Despite the inspiring prospects and results of user identity linkage, there are several key challenges that remain to be solved:

Heterogeneity nature of mobility data: Due to the different usage behavior of users and various collection mechanisms, the properties of mobility data (*e.g.*, sample rate, time periods) are drastically different across services. For example, the mobility traces

[1] https://github.com/vonfeng/DPLink

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313424

collected by an Internet Service Provider (ISP) are over 3 times denser than those collected by an online social network. Early works simply assume mobility traces of different services have similar sample rate [20] or time period [24], which are sensitive to the heterogeneous mobility data and do not perform well in practice. Due to this challenge, the general trajectory similarity [17, 21] algorithm designed for the trajectory from the same data source also fails to model the correlated relationship between the different data source and can not be effectively applied in the linkage problem.

Poor quality of mobility data: The data quality of collected mobility data is not always so good. *On the one hand*, the collected data only records the time and location information of mobility which is not enough to mine the hidden semantics of it. *On the other hand*, due to the limitation of devices and other artificial reasons [31, 36], the collected data usually contains noisy records to generate significant spatial and temporal mismatches between trajectories from different service. Because of the limitation of algorithms and the lack of proper data, existing approaches [24, 25] ignore the hidden semantics of mobility trajectory. Although, some works [32] propose prior knowledge-based solutions to address the mismatch problem. These solutions require proper manual parameter settings and are difficult to be applied in reality.

In this paper, we propose *DPLink*, an end-to-end deep learning based framework, to achieve linking IDs belonging to the same user for mobility data collected from different services with heterogeneous nature. *DPLink* is consist of two main components: *feature extractor module* and *comparator module*. The *feature extractor module* is designed to extract vector features from input raw trajectory and model relations between the trajectories from the different data source. The following *comparator module*, implemented as a multi-layer feed-forward network, is aimed to yield the final similarity score of the extracted representative trajectory feature vectors.

As the core component of *DPLink*, *feature extractor* contains two level encoders: *location encoder* and *trajectory encoder*. The *location encoder* is designed to integrate multi-dimensional input and extract the low-level feature of isolated locations. The multi-modal embedding based design in location encoder makes *DPLink* expandable to other available geographical features like PoI context. The following *trajectory encoder* is designed to capture the transitional relations of a single trajectory itself and model the correlation between two different trajectories. In the *trajectory encoder*, the transitional relations of single trajectory are captured by a *recurrent encoder*. Then a *selector* is introduced at the end of *trajectory encoder* to force the model to focus on the discriminative parts of trajectory and model the correlations between two different trajectories. With the help of this attention based selector, *DPLink* not only is able to observe the similar parts of two trajectories but also works robustly with noisy and missing trajectory.

Besides, we propose a pre-training mechanism to address the training challenges introduced by the heterogeneity nature of mobility data. Following our proposed training mechanism, our model is first trained to complete a warm-up task as linking trajectory from one mobility data but with different time periods. Due to the regularity of human mobility [10, 27] and the consistency and high quality of used mobility data, this single data based linkage task is much easier for *DPLink* to complete than the linkage task on cross-domain datasets with different quality. Intuitively, this single data based linkage task acts as a simple *auxiliary task* to help the model to first learn about the basic knowledge of the trajectory space. Then, the pretrained network with prior knowledge of the physical world and trajectory data is trained to complete the final *target task*: user identity linkage task on different data collected from different services.

Our contributions can be summarized as follows:

- We are the first to use deep learning techniques in the user identity linkage problem based on the heterogeneous mobility traces collected from different services with different quality. Our model does not assume any property of mobility data and works with heterogeneous mobility data across services.
- We utilize recurrent network with pooling unit as a trajectory encoder to extract transitional features from each single trajectory and introduce attention based selector to capture potential correlations between trajectories from two different data source. With fusing these two features together, our model obtains a comprehensive high-level understanding of trajectory.
- We propose a simple yet effective pre-training mechanism to adapt to the heterogeneity nature of different source mobility data. And the proposed training mechanism greatly improves the performance and robustness of our deep learning model on user identity linkage task on the different data source.
- We perform experiments on two real-life mobility datasets and compare the performance of our model with eight baselines. Extensive results demonstrate that *DPLink* outperforms the state-of-the-art solutions by more than 15%. Moreover, our model succeeds in utilizing additional geographical context data to further improve the performance and works robustly with noisy data.

The rest of this paper is organized as follows. We first formulate the problem in Section 2. Following the formulation, in Section 3, we describe the details of the whole framework of *DPLink*. In Section 4, we apply our model to two real-world mobility datasets and conduct extensive experiments. After systematically reviewing the related works in Section 5, we finally conclude our paper in Section 6.

2 PROBLEM FORMULATION

Let $\mathcal{A}$ represents the set of online account IDs, and S represents the set of different online platforms. Then, $\forall s \in S$, $\mathcal{A}^s$ denotes the set of online account IDs on platform s. Let a tuple $p = (l, t, e)$ represents a location record in location l with semantic label e at time t. Note that locations and times may be recorded at a different granularity and levels of precision in different source mobility data (*e.g.*, GPS coordinates or nearest base station in location record). Thus, without loss of generality, locations and times are divided into bins corresponding to geographical regions (*e.g.*, street block) and intervals of time (*e.g.*, one hour). In this paper, semantic label e is the distribution of point of interests (PoIs). But, we demonstrate that it is not limited to the PoI context, other information like textual context from user review can also be easily applied to our model without too much modification of network. We further define $\mathcal{T}$ as the set of all time bins, $\mathcal{L}$ as the set of all locations, and $\mathcal{E}$ as the set of PoIs.

Given any online ID $u \in \mathcal{A}$, we define its location records as, $\mathcal{R}(u) = \{p_1, p_2, ..., p_n\}$, where n denotes the number of location

Table 1: A list of commonly used notations.

Notat.	Description
$\mathcal{A}$	The set of all online IDs.
S	The set of types (platform) for online IDs.
$\mathcal{A}^s$	The set of online account IDs on platform s.
$\mathcal{T}$	The set of all time slots.
t	time stamp.
$\mathcal{L}$	The set of all regions.
l	location indicator.
$\mathcal{E}$	The set of PoIs.
e	semantic information.
$p = (l, t, e)$	A location record.
x	The embedded location record.
h	The hidden state of recurrent network.
u, v	online user ID.
$\mathcal{R}(u)$	location records sequence for online ID u.
$r(u)$	trajectory slice for online ID u.
$I(u, v)$	Binary variable indicating whether ID u and v belong to the same users.
N	The number of linking candidates for online ID u.

records. Taking the continuity of mobility data into consideration, we further partition the location records into meaningful trajectories $\mathcal{R}(u) = \{r_1(u), r_2(u), \ldots\}$ with maximum time window T_w (*e.g.*, 1 day). Given a pair of online IDs $u \in \mathcal{A}^1$ and $v \in \mathcal{A}^2$, let a binary variable $I(u, v)$ indicates whether these two IDs belong to the same user when the trajectories of them $r(u), r(v)$ are known,

$$I(u,v) = \begin{cases} 1, & u,v \text{ belong to the same user.} \\ 0, & u,v \text{ belong to different user.} \end{cases}$$

Further, given a target ID u, a list of candidates IDs $v_1, v_2, \ldots, v_N \subseteq \mathcal{A}^2$ and their trajectories $r(u)$ and $r(v_i)$ for $i = 1, 2, \ldots, N$, we aim to build a function, which is approximate to the identity function I enough, to find the best matching trajectory $r(v_i)$ and ID v_i.

Following the aforementioned definitions, many researchers [24, 25, 32] proposed insightful algorithms to work out this problem. However, because of the data quality problem and heterogeneity of mobility trajectory, these methods are still far from application in reality. Inspired by the powerful representation ability of deep learning models, we propose a deep learning based framework to address these challenges and aim to achieve better performance on real-life mobility dataset. The difference in workflows between existing works and our work is shown in Figure 1. In the traditional workflow of existing works, they usually first align the trajectory from the spatial or probabilistic view and then calculate the spatial or distribution distance to obtain a score. However, instead of aligning trajectory directly, we aim to utilize deep learning tool to first obtain the latent representation of mobility trajectory and then calculate the similarity between these two vectors as the similarity of trajectories. The details of our proposed model can be found in the next section.

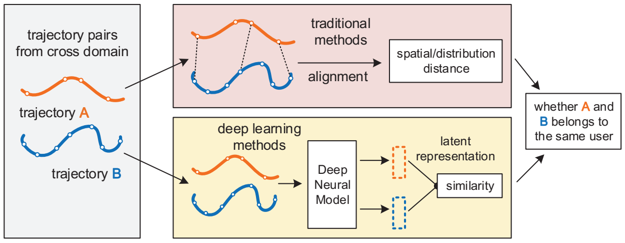


Figure 1: Different working mechanism of traditional methods and our proposed model.

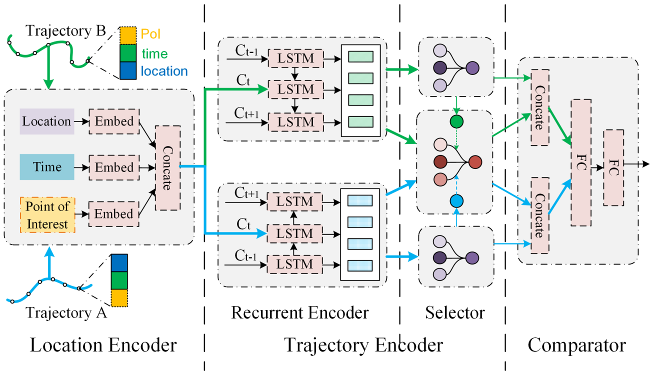


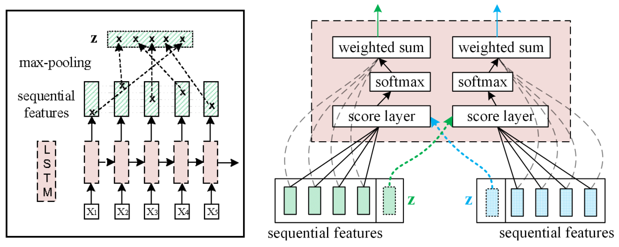
Figure 2: Main architecture of *DPLink* consisting of three major components: location encoder, trajectory encoder (including recurrent encoder and attention based selector), and comparator network.

3 MODEL AND METHOD

The structure of *DPLink* is presented in Figure 2. *DPLink* contains three major components: *location encoder*, *trajectory encoder*, and *comparator network*. The input trajectory pairs are first processed by *location encoder* and then fed into *trajectory encoder* to extract multi-level representative features. Particularly, *DPLink* uses a attention based *selector* at the end of *trajectory encoder* to focus on the similar parts of the trajectory pair and avoid the potentially harmful influence of missing and noisy data. Finally, the generated vector pair, which represents the original trajectory pair, are fed into the *comparator network* to calculate the similarity score of two input trajectories.

3.1 Location Encoder

Location encoder is a multi-modal embedding module, which is designed to reorganize and embed the spatial and temporal features of a trajectory point $p_i = (t_i, l_i)$ into a single vector x_i. Further, due to the flexibility of embedding network, we can also easily model external information e_i like geographical context information in the embedding network. In this way, our model is able to capture the hidden semantic behind the spatial-temporal point, which is difficult to utilize and model for existing methods. We design three sparse linear embedding layers to encode each type of input (*e.g.*, one-hot) into a dense vector representation. Then, we concatenate them together to obtain an ensemble vector x_i. To strengthen the modeling ability of the embedding module, we add *tanh* function



(a) Recurrent encoder.

(b) Attention-based selector.

Figure 3: Details of trajectory encoder in *DPLink*.

as the final non-linear activation function. The formulation of the embedding module is as follows,

$$x_i = tanh(W_p p_i + b_p) = tanh([W_t t_i + b_t; W_l l_i + b_l; W_e e_i + b_e]),$$

where W and b denote the learnable parameters of embedding layers, $tanh$ denotes the non-linear activation function, $[;;]$ denotes the concatenate function.

As mentioned above, the location fed into our model is in the form of a one-hot vector. Compared with the original geographic coordinates, one-hot vector lose the information of spatial dependencies, which means that regions close to each other in the physical world can be far away in the one-hot space. However, the spatial dependency is important in measuring trajectory similarity. In order to enable *DPLink* to learn about the spatial adjacency of location, we introduce mobility prediction task to help our model learn the meaning of closeness of location. This task is to predict the next location of an object with knowing its trajectory history. Due to the regularity of mobility, the location points of the trajectory which neared in the time dimension is also neared in the spatial dimension. In other words, the location encoder is enforced by the mobility prediction task to embed the adjacent location in the physical world into the adjacent space in the latent high-dimensional space in a neural network.

It is noted that this multi-modal embedding module is shared by the network to simultaneously process two input trajectories. This sharing mechanism guarantees that two types trajectories from the same geographical space can be projected into another same latent space. Besides, the shared embedding module also greatly reduces the parameters of the whole network.

3.2 Trajectory Encoder

Following the *location encoder* is the *trajectory encoder*, which contains a *recurrent encoder* to extract the transitional level feature of single trajectory and a *selector* to extract correlated level features of trajectory pair.

3.2.1 Recurrent Encoder. Based on the embedded vector representation $\{x_1, x_2, ..., x_n\}$ of original trajectory $\{p_1, p_2, ..., p_n\}$, recurrent encoder is introduced to capture the sequential transitions and model the mobility pattern of a single trajectory. The output encoded trajectory feature is recorded as vector representations $\{h_1, h_2, ..., h_n\}$. As Figure 3(a) shows, the recurrent encoder is made up of a recurrent neural network with max/mean pooling operation. Recurrent neural network is a standard class of neural networks, which is designed with cycle and internal memory to model the sequential information. We use the widely used long short term memory (LSTM) [11] and its popular variation gated recurrent unit (GRU) [5] as the basic unit of our recurrent encoder.

To enhance the ability of sequential pattern modeling, we apply stacked techniques in the recurrent encoder. In practice, we stack several recurrent layers by feeding the output of the former recurrent layer into the next recurrent layer. To overcome the potential overfitting introduced by too many recurrent units, we apply dropout to the output of every recurrent layer.

Since the number of features extracted from the recurrent network is still identical to the length of trajectory, we introduce pooling operation (*e.g.*, max pooling and mean pooling) after recurrent network to obtain fix-length vector representation of trajectory. Actually, this operation is beyond only feature reshaping but also achieves a certain range of performance improvement in our experiment. This is because that pooling operation acts as a simple self-feature filter to select the important features. As Figure 3(a) shows, with the help of recurrent network with max pooling, we model the sequential relations $\{h_1, h_2, ..., h_n\}$ and finally obtain a single feature vector z for each trajectory. And we call this single feature vector z as the *primary representation vector* for trajectory.

3.2.2 Co-Attention based Selector. To handle the data quality problem and enable our model to focus on the critical parts of the trajectory pair for matching and linking, we propose to design a co-attention based selector network. We first briefly introduce the background knowledge of attention mechanism.

Given a query vector q and a series of candidate vectors $\{h_1, h_2, ..., h_n\}$, attention mechanism can be implemented with two steps: 1) to calculate the “correlation” between the query vector q and all these candidate vectors $\{h_1, h_2, ..., h_n\}$; 2) with these normalized “correlation” as weights $\{\alpha_1, \alpha_2, ..., \alpha_n\}$, to calculate the weighted sum y of candidate vectors as the comprehensive representation of them. This weighted sum y is regarded as the summary of the most related parts of candidates $\{h_1, h_2, ..., h_n\}$ for the query q. There are three widely used attention methods: *dot*, *general*, *mlp*. The main difference between these attention implementations is the calculation of “correlation”. The formulation of typical attention methods are as follows,

$$y = \sum \alpha_i h_i, \quad \alpha_i = \sigma(f(q, h_i)),$$
$$f_{dot}(q, h_i) = h_i^T q,$$
$$f_{gen}(q, h_i) = h_i^T W q,$$
$$f_{mlp}(q, h_i) = v^T tanh(Wq + Uh_i),$$

where W, U, v are the learnable parameters, f represents the score function, σ is the soft-max function, h_i represents i_{th} candidate vector, q is the query vector and y is the final output.

Figure 3(b) presents the core idea of the co-attention based selector. In the common application of attention mechanism like neural machine translation, there is only one candidate sequence, where the query vector q of attention is naturally constructed from candidates $\{h_1, h_2, ..., h_n\}$, *e.g.*, $q = h_n$. Different from the former, we introduce co-attention mechanism to adapt general attention network for pair input to capture and model the correlation relationship. For the candidates $\{h_1^A, h_2^A, ..., h_n^A\}$ from trajectory A, we use the *primary representation vector* z_B of trajectory B as the query vector $q_A = z_B$. Meanwhile, we use the *primary representation vector* z_A of trajectory A as the query vector $q_B = z_A$ for the candidates $\{h_1^B, h_2^B, ..., h_m^B\}$ from trajectory B. In this way, we directly connect two trajectories before the final comparator network and give them opportunities to find the related parts of each other. Further, this co-attention based selector will reduce the harmful effects from potential noisy records. After the processing of co-attention network, we obtain the final feature vector of trajectory: y_A for original trajectory A $\{p_1^A, p_2^A, ..., p_n^A\}$ and y_B for original trajectory B $\{p_1^B, p_2^B, ..., p_m^B\}$.

3.3 Comparator Network

The final component of our model is a comparator network, which is implemented as a multilayer feed-forward network. The final layer of the comparator network is a neural unit with *sigmoid* function acting as a logistic regression function to generate the final similarity score. The feature vectors (y_A, z_A) and (y_B, z_B) are independently fused together to obtain the final vector presentation of each trajectory. Then these two vector representations are fed into multi-layer feed-forward network to yield the similarity *score*. From the view of the problem, the comparator network gives two input trajectories a second chance to exchange information and validate each other. From the view of the network, the comparator network can be regarded as a simple but effective classifier for binary classification to judge whether two input vectors belong to the same class. Due to the *sigmoid* function, the output score is a normalized probability which can be optimized by binary cross entropy loss function.

3.4 Training strategy

Our model works in an end-to-end manner without requiring handcrafting features. Since we translate the user identity task into a binary classification problem, we choose binary cross entropy loss as the objective loss function.

$$Loss = -\sum_{i=1}^{n} y_i \log x_i + (1 - y_i) \log(1 - x_i),$$

In the training, the Adaptive Moment Estimation (Adam) algorithm is utilized to optimize the model. Several widely used tricks, such as dropout, L2 regularization, and learning rate schedule, are used to avoid the overfitting problem. The detailed settings for these parameters can refer to the next section.

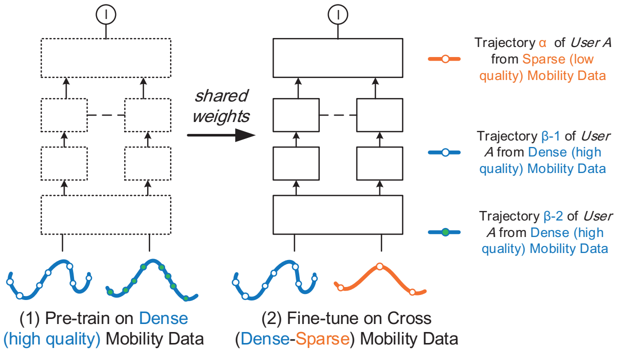

shared weights
Trajectory α of User A from Sparse (low quality) Mobility Data
Trajectory β-1 of User A from Dense (high quality) Mobility Data
Trajectory β-2 of User A from Dense (high quality) Mobility Data
(1) Pre-train on Dense (high quality) Mobility Data
(2) Fine-tune on Cross (Dense-Sparse) Mobility Data

Figure 4: Two step training strategy for user identity linkage task.

However, due to the heterogeneous nature of mobility trajectories from the different service and platform, our model always fails to converge and stops with really poor performance when directly trained from scratch. The intuition that mobility trajectories from different platforms share identical underlying spatial-temporal pattern is the precondition for completing the user identity linkage task. Nevertheless, due to the heterogeneous nature of mobility trajectories from different data source, our model struggles to explore the huge spaces and learn the true knowledge of it. On the one hand, different platforms provide users with various services and users generate different mobility behaviors with various intentions. On the other hand, the sample rate and recording mechanism of collecting data in different platforms are also different. All of these lead to the heterogeneous nature of mobility trajectories, which make it difficult for our model to learn valuable knowledge. Consequently, our original model fails to link trajectories across different mobility data.

Here, we propose to pretrain the whole model with a simple task as warm-up to first obtain the basic knowledge of the physical world and trajectory pattern. Once our model achieves good performance on this simple task, we start to train it with the final user identity linkage task on different mobility data source. This warm-up task is to identify whether two mobility trajectories from the same platform in different periods (*e.g.*, different days) belong to the same user. This warm-up task is proposed based on the observation from the regularity of human mobility [10, 27], people usually go to the same places (*e.g.*, to and from where he needs to go like home and office) and generate similar even the same trajectories in different work days. Furthermore, the trajectory used in the warm-up task are both from the identical high-quality mobility dataset, which is denser and without too many missing records. Due to the regularity of mobility and the high-quality mobility dataset, the warm-up task becomes an easier and directional task to help the model to converge and learn the useful prior knowledge for the future difficult task.

The two-step training strategy is presented in Figure 4. Our training strategy can be regarded as a kind of network-based transfer learning method. The prior knowledge learned from the first linkage task on a single dataset is transferred to the final linkage task (*target task*) on different datasets by sharing partial network weights.

4 PERFORMANCE EVALUATION

In this section, we conduct extensive experiments on two real-world datasets to answer the following research questions:

- **RQ1:** How does *DPLink* perform in the user identity linkage problem on real-life datasets compared with existing algorithms?
- **RQ2:** How does the proposed training mechanism help to improve the performance of *DPLink* on the user identity linkage task?

Table 2: Statistics of the datasets.

Dataset	Users	Records	Locations	Rec./U.	Loc./U.	Duration
Twitter	1228	53337	8975	43	25	21 months
Foursquare	2970	44915	8975	15	12	48 months
ISP	2844	325215	12576	114	19	1 week
Weibo	1761	49651	12576	28	5	1 week

- **RQ3:** Can *DPLink* be applied in data with poor quality (*e.g.*, noisy records)? Is it possible to utilize more external geographical context data (*e.g.*, PoI information) to improve the performance?

In what follows, we first describe the experimental settings, and then answer the above three research questions one by one.

4.1 Experimental Settings

4.1.1 Datasets. We carry out experiments with two real-world cross-domain mobility datasets. The first dataset is the mobile network record dataset provided by one of the largest major Internet Service Provider (ISP) in China and the social network location service records obtained from Weibo[2]. The second dataset, provided by Zhang et al. [41], is the location check-in records from Foursquare[3] and Twitter[4] with the same users. Different generation and collection mechanism of these datasets ensure the representativeness of our experiments. Table 2 summarizes the basic statistical information of datasets. Below, we describe these datasets in detail.

ISP-Weibo. The ISP dataset contains 325215 mobility records that cover the metropolitan area of Shanghai from April 19 to April 26 in 2016. The location records are generated in the base station level when users access cellular network via mobile devices for communication and the Internet. Each trajectory is characterized by anonymized user ID. It contains a series of spatial-temporal points produced by the user, where each point includes a base station ID and a timestamp. We replace the base station ID with a certain longitude-latitude coordinate according to the base station information provided by ISP. The Weibo dataset is generated from ISP dataset. Our collaborators from ISP collected the Weibo sessions from the ISP datasets in the same time window with the permission of Weibo. In Weibo dataset, each mobile trace is characterized by a Weibo ID and contains a series of GPS coordinates that show up in HTTP sessions between the mobile application and the Weibo server. These GPS coordinates are produced when users access the location service in their mobile applications like check-in. Before we access the dataset, the collaborators have mapped the Weibo ID into the same anonymized user ID in the ISP dataset to protect the personal privacy.

Foursquare-Twitter. This dataset contains the trajectories data from Foursqaure, a popular location-based social network (LBSN), and Twitter, a micro-blogs social network around the world. As its primary function, Foursquare provides users various location-related services like location check-ins and posting online reviews. Twitter also provides users with basic location-based services like location check-ins. Provided by Zhang et al. [41], the data is crawled from the web pages of thousands of users who use both platforms with at least one location records during November 2012 in the east coast of U.S.. Based on the crawled raw data, the trajectory is also characterized as a user's account ID and a series of GPS coordinates with timestamp.

Table 3: PoIs category distribution on ISP-Weibo dataset.

Related Function	PoI Categories
Residence	residence, life services.
Entertainment	food, hotel, gym, shopping, leisure.
Business	finance, office building, company, trading area.
Industry	factory, industrial estate, economic development zone.
Education	school, campus.
Scenery spot	scenery spot, tourism development.
Suburb	villages, towns.

Table 4: Typical parameter settings.

Item	Value	Features	input	output
batch size	32	location	10000	200
learning rate	1e-3	time	24	10
drop out	0.3	PoI	20	10
fine-tune lr	0.0005	hidden state	200	200

We crawled 0.75 million point of interests (PoI) of Shanghai from BaiduMap[5] as the additional geographical context of ISP-Weibo dataset. As Table 3 shows, the crawled PoI dataset contains 20 categories and can be classified into 7 region functions. For every base station in ISP dataset, we calculate the distribution of PoIs for it in the surrounding 1 kilometer area. This PoIs distribution serves as a soft function label for the region and describes the potential intention of the user's movement.

It's noted that all the user IDs in our datasets are anonymized to protect user's privacy. Meanwhile, we store the data in a secure local server and only core researchers can access the data with strict non-disclosure agreements.

4.1.2 Baselines. We compare the performance of our model with eight state-of-the-art baselines. Six of them are classic user identity linkage algorithms, another two are deep learning based trajectory representation models.

NFLX: With knowing some external information, Narayanan et al. [23] propose a statistical model based mobility trace similarity score to identify the users in Netflix dataset. NFLX cannot be directly applied into cross-domain linkage problem and we follow the method [24] to adapt it to our problem.

MSQ: Ma et al. [20] incorporate general knowledge in forms of global movement constraints and preferences to identify users from dataset. Specifically, they consider negative square difference between two mobility traces as their similarity score.

[2] https://weibo.com

[3] https://www.foursquare.com/

[4] https://www.twitter.com

[5] https://map.baidu.com/

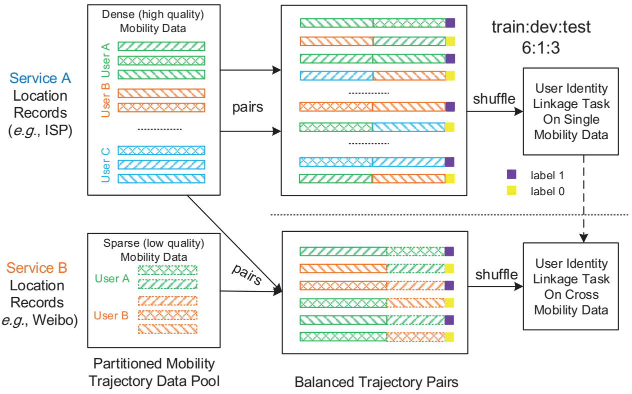


Figure 5: Build training/validation/testing dataset.

HIST: Naini et al. [22] focus on linking users by matching the location histograms of their mobility traces. Firstly, they compute a user's visiting frequency of each location and then define a similarity score of two histograms based on Kullback-Leibler divergence.

LRCF: Goga et al. [8] take the popularity of different locations into consideration. LRCF applies the term frequency-inverse document frequency(TF-IDF) [28] weighting scheme to location visiting histograms and measures mobility trace similarity using a cosine distance.

WYCI: Rossi et al. [25] propose a time based probabilistic user identifying algorithm. They use the frequency of user login in different locations to approximate the probability of visiting these locations. Then they determine if a mobility trace belongs to a user by computing the user's probability to produce the mobility trace.

POIS: POIS [24] algorithm uses the "encountering" events to measure the similarity of two different-domain mobility traces. It assigns every "encountering" event a weight based on statistical model and uses the weighted sum as the similarity measure.

As we are the first deep learning based model for user identity linkage task on different mobility datasets, we compare our model with two deep learning based models for trajectory representation.

TULER: Gao et al. [7] utilize the recurrent network to encode the trajectory into a single vector to identify users in single mobility data. TULER cannot be directly applied to our problem, thus we train two independent TULER models for each mobility data and check whether these two models can identify each input trajectory as the same user.

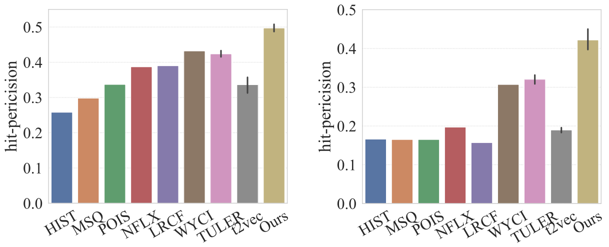
t2vec: Li et al. [17] propose to adapt seq2seq model with a spatial proximity aware loss function to infer and represent the underlying route information of a trajectory for efficient trajectory similarity computation. Following the philosophy of original paper, we adapt it to our problem by using sparse (low sampling) trajectory from one dataset as input and using dense (high sampling) trajectory with same user ID from another dataset as the underlying route to learn the representation of trajectory. Then, we use this representation to link different IDs.

4.1.3 Preprocessing. Figure 5 presents the details of how to build training/validation/testing dataset in our experiment. As Figure 5 shows, the first step is to divide the lengthy location records into continuous mobility trajectories. After this operation, we obtain two independent mobility trajectory pools, where each trajectory is labeled with its user ID, for two different mobility data. As mentioned in the training section, we first pretrain the model with user identity linkage task on single data as a warm-up and then fine-tune the model with user identity linkage task on cross-domain mobility data. Furthermore, we choose the high-quality mobility data with more frequent and longer records, which is called *dense* mobility trajectory data to complete the warm-up linkage task. Another mobility data source with fewer mobility records called *sparse* mobility data is only used in target user linkage task on cross-domain datasets. The mechanism of building data in two training stages are the same, while the main difference is to choose which data to use.

We first introduce the core step of building experiment data for user identity linkage task on single domain data as follows. We randomly choose two trajectories from the dense mobility data pool with identical owner labels to build positive trajectory pair. The basic intuition behind this operation is the regularity of human mobility [10, 27], which means that the mobility trajectory of one user keeps similar in the different periods with extremely high probability. To meet the sample balance requirement of binary classification problem in training stage, for every positive trajectory pair we randomly select two trajectories from dense mobility data pool with different user IDs to form a negative trajectory pair. In the training stage and validation stage, only this one negative sample is enough.

However, in reality, we usually have lots of online ID candidates for matching, and we need to compare them together to find the best matching. To simulate this practical case, in the testing stage, we choose N negative candidates for matching. Particularly, for each online ID, we choose those left online IDs whose trajectory have at least one intersection with it as candidates. Based on the statistics of our data, the average number of these candidates for each online ID is around 20. To meet the requirement of batch size for efficient computation, we select 32 as the default setting for N in the experiment. To build experimental data for user identity linkage task from different data sources (*target task*), we follow the similar steps mentioned before. Different from choosing two input trajectories from the same mobility dataset for warm-up linkage task, we choose one trajectory from dense mobility data and another trajectory from sparse mobility data for final user identity linkage task. In our experiment, the proportion of training, validation and testing data is 6:1:3. It's noted that we divide not only the trajectory but also the users by shuffling the trajectory data with user label. This operation makes sure the generalization of our experiment, where some users may only have few data in the training step and even they only appear in the testing step.

4.1.4 Metrics and Parameter Settings. Our model is implemented on Pytorch platform and the typical parameter settings of our model are presented in Table 4. In the experiments, we select GRU as the default recurrent unit and *dot* attention as the default attention mechanism for efficient computation. The results for LSTM and other basic components are similar. Following the aforementioned evaluation dataset, our model is trained and validated in a binary classification manner, and finally evaluated in the search and ranking manner.



(a) @ISP-Weibo dataset.

(b) @Foursquare-Twitter dataset.

Figure 6: The performance of our model and baselines on two real-life datasets.

In the training step, we use the F1 score and AUC to measure the performance of our model, which are both widely used metrics in classification problems. The F1 score is defined as,

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}.$$

AUC, known as the area under the receiver operating characteristic (ROC) curve, is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. With P positive instances and Q negative instances, AUC can be calculated by the following formulation,

$$AUC = \frac{\sum_{i \in +} rank_i - P(1+P)/2}{P \times Q}.$$

In the validation step, we also use the F1 score and AUC as the metrics to cooperate with the early stopping mechanism to find the best parameters and avoid overfitting.

In the testing step, we use hit-precision as the metric to evaluate the performance of algorithms in the user identity linkage task with more than one (*e.g.*, $N = 32$, as mentioned in the preprocessing section) candidate online IDs. After *DPLink* deciding the score for each candidate, we rank these candidates by score and calculate the hit-precision. Hit-precision of top-k candidates is defined as follows,

$$h(x) = \begin{cases} \frac{k-x}{k}, & \text{if } k > x >= 0 \\ 0, & \text{if } x >= k \end{cases}$$

We choose $k = 5$ as the default setting for hit-precision.

4.2 Overall performance on user identity linkage task. (RQ1)

We evaluate our model with eight baselines on ISP-Weibo and Foursquare-Twitter datasets to present the performance on user identity linkage task. The results are presented in Figure 6.

We first analyze the results of ISP-Weibo dataset. Among six classic linkage baselines, WYCI algorithm with the tolerance of different sample rate performs best with 0.432 hit-precision. Besides, the performance of NFLX and LRCF with the tolerance of mismatch from the spatial or temporal view are also similar well. Particularly, the hit-precision of our model is 0.497, which is 15% higher than the best baseline WYCI.

Besides, we compare our model *DPLink* with other two state-of-the-art trajectory representation algorithms: TULER and t2vec. The results demonstrate that our model outperforms both of them more than 17%. TULER is designed for directly identifying the user from single mobility dataset. It fails to obtain better performance due to two potential reasons: 1) TULER cannot identify any new users because the user set is decided in the training step. However, in our experiments, the users in the testing step is not identical to the training step; 2) each TULER only focus on its single mobility dataset and ignores the potential correlation between two different mobility dataset. Based on seq2seq model, t2vec is proposed to infer and represent the underlying route information of a trajectory for efficient trajectory similarity computation. In practice, t2vec needs extremely high-quality dataset (*e.g.*, taxi trajectory dataset with less than 1 minute sampling period) to learn good enough representative function. It fails to handle with the poor quality (low sampling and missing records) and the heterogeneity nature of our cross-domain mobility dataset. *DPLink* is designed for the user identity linkage task and succeeds in dealing with these challenges with special network design and training strategy. On the one hand, the location encoder and trajectory encoder help *DPLink* extract representative features from a single trajectory. On the other hand, the attention based selector enables *DPLink* to capture the correlation between two trajectories with different properties from different mobility data. Finally, the MLP based comparator network acts as a powerful classifier to obtain the final result.

The similar results can also be found in the Foursquare-Twitter dataset. Because of the smaller volume and sparse nature of Foursquare-Twitter dataset, the performance of all the methods is lower. However, due to the powerful representation ability of neural network and transfer learning based training strategy, our model still achieves about 0.4 hit-precision. Compared with the-state-of-the-art methods, the performance gain of our model can be up to 25%. In a word, the evaluation results of two real-life datasets demonstrate the superiority of our model than other the-state-of-the-art methods.

4.3 The effects of the training strategy. (RQ2)

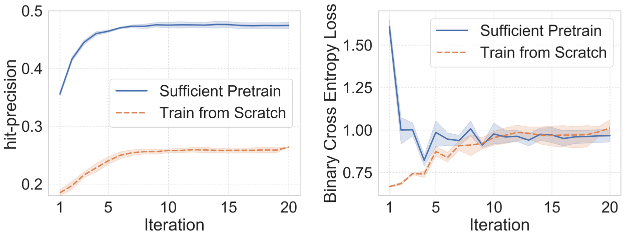
In this section, we conduct experiments to demonstrate the effect of pre-training strategy for *DPLink*. We take the results on ISP-Weibo dataset for example.

Table 5 shows the performance of our model with different training strategies on ISP-Weibo dataset. Obviously, the pretrain step is crucial for better performance. Furthermore, we choose to remove different components of *DPLink* from the sharing weights step to locate the most important component for the performance. As Table 5 shows, removing location embedding and selector weights from the sharing step leads to the worst performance, which denotes the crucial role of location encoder and attention-based selector in modeling. Besides, the effect of removing the recurrent encoder is smaller and the removing the comparator network weights even does not harm the performance at all. These little effects tell us that the transitional relations and classification knowledge are not hard to learn from beginning in this task.

We dive into the training process to understand the importance of pretrain step. As Figure 7(a) shows, after pretraining, the initial performance of our model before fine-tuning on target task can

Table 5: Performance of different training mechanisms on ISP-Weibo dataset. For example, "Full−Location Encoder" denotes that the embedding weights of location encoder is not shared in the pretrain step, which means the location encoder is trained from scratch in user identity linkage task.

Training Strategy	hit-precision (mean+std)	Δ
Train from Scratch	0.257 ± 0.025	-48.3%
Full Pretrain	0.497 ± 0.013	0
Full−Location Encoder	0.404 ± 0.018	-18.7%
Full−Recurrent Encoder	0.438 ± 0.005	-11.9%
Full−Selector	0.395 ± 0.014	-20.5%
Full−Comparator	0.499 ± 0.009	+0.4%



(a) The testing performance curves.

(b) The testing loss curves.

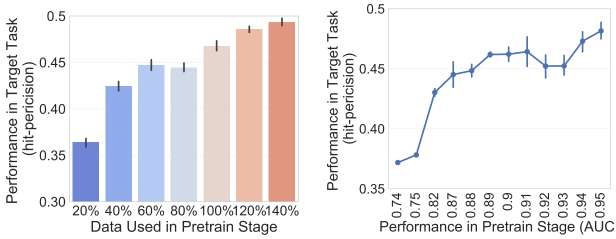
Figure 7: The comparison results between two types training mechanisms on ISP-Weibo dataset.

be higher than 0.35, which is better than the best performance of model trained from scratch. With fine-tuning on target task, the hit-precision of our model can be further improved to 0.49. Figure 7(b) presents the variation trend of testing loss of models with different training mechanism. Due to the heterogeneity of mobility data, the model trained from scratch becomes over-fitted at the very start. Meanwhile, the pretrained model really learns the knowledge about the task and performs better and better.

Figure 8 presents the effects of data volume and model performance of pretrain step on the performance of our model on target task. As Figure 8(a) shows, with more and more data utilized on single domain linkage task in the pretrain step, the final performance of our model on target task also becomes better and better. This observation tells us that enough unlabeled data on a single domain can help to improve the performance of model on target task and reduce the requirement of labeled cross-domain data. Figure 8(b) shows us that better pretrained model on single domain linkage task produces better fine-tune model on target task.

4.4 The expansion capability and stability of *DPLink* model. (RQ3)

In this section, we evaluate whether *DPLink* can utilize external information with the help of multi-modal embedding based location encoder to improve the performance on user identity linkage task and whether it works robustly with noisy records.



(a) The effect of data volume used in pretrain stage.

(b) The effect of the performance in pretrain stage.

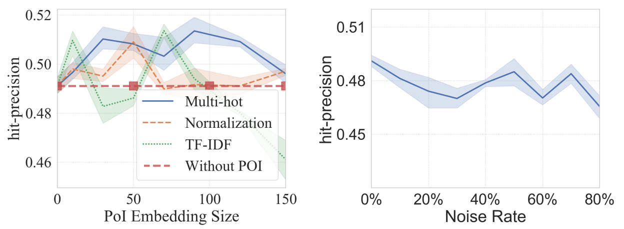
Figure 8: The effects of different pertrain parameters on ISP-Weibo dataset.

To evaluate the expansion capability of multi-modal embedding network, we build three kinds of PoI distribution features: 1) multi-hot distribution, which only describes whether specific type PoI exists in the region; 2) normalized probability distribution of different PoIs in the region; 3) tf-idf distribution, the enhanced version of normalized probability distribution. The evaluation results are presented in Figure 9(a). The results show that *DPLink* is able to apply external PoI information to further improve the performance from 0.49 to more than 0.51. Particularly, the multi-hot feature of PoI distribution performs much better and stable. However, *DPLink* is not limited to only use this kind of PoI distribution feature. Other external information like textual information can also be directly applied to our model after simple pre-processing like word2vec operation.

Figure 9(b) presents the performance variations of *DPLink* with noisy trajectory input. In Figure 9(b), noise rate in 20% means that 20% trajectories in the data are randomly selected to randomly replace one real data point in it with a noisy point. As Figure 9(b) shows, the performance of our model keeps stable with the increase of noise rate, which demonstrates the robustness of *DPLink* . Furthermore, we visualize the attention weights of three kinds of points in the trajectory to verify whether our co-attention based selector really plays an import role in filtering out the harmful effects of noisy records. Three types points are 1) intersection location between two trajectories; 2) noisy point inserted by the program; 3) other location points. In Figure 10, we analyze the average attention weights of three types points in 4 scenarios which include the dense/sparse (high/low quality) mobility data with positive/negative labels. As Figure 10 shows, the average attention weight for intersection point is the highest and the weight for the noisy point is the lowest, which distinctly demonstrates the selecting value of our co-attention based selector.

5 RELATED WORK

Applications of Identity Linkage: A number of applications can benefit from linking IDs across services. For example, Kumar et al. [16] investigated the user migration patterns across social media to provide guidance for online social network design. Zafarani et al. [40] and Yan et al. [37] leveraged linked IDs across social networks for better friend recommendations. Yang et al. [38] leveraged linked IDs across sites for better video recommendations.



(a) The effects of geographical context with different PoI embedding size.

(b) The effects of noisy records with different noise level.

Figure 9: The effects of data quality (geographical context and noisy records) on our model.

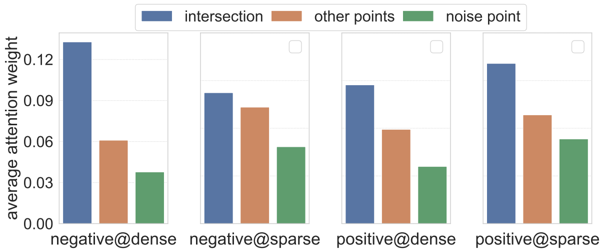


Figure 10: Average attention wights of three types points on testing dataset. For example, "negative@dense" means that the attention weights are from the dense(high quality) mobility trajectory with negative label.

All these works indicate the strong motivations for the service provider to link IDs belonging to the same user. Besides, some researchers [30, 35] tried to protect users from identification attack by matching mobility trajectory.

Identity Linkage using Trajectory Data: A few recent works examine the possibility of linking IDs based on location data [2–4, 13, 20, 22–26, 29, 33, 34]. Mudhakar et al. [29] and Ji et al. [12, 13] focused on linking IDs based on users' graph/network structures. They adapted their algorithms to location trajectories by constructing a "contact graph" to model users encountering with each other. However, these algorithms still require using social network graphs, which are not available in our scenario. In addition, some algorithms are designed to tolerate data noise such as temporal mismatching [23] and spatial mismatching [20]. Wang et al. [32] proposed algorithms with tolerating spatial or temporal mismatches (or both) and modeling user behavior for better linkage performance. Other algorithms implemented de-anonymization attacks based on *individual user's* mobility patterns [22, 25, 26]. Finally, researchers also developed identity linkage algorithms based on "encountering" events [2–4, 24]. By considering the location context (*e.g.*, user population density), it achieved a better performance [4, 24]. However, none of them is able to capture the trajectory dynamics and integrate semantic information simultaneously.

Representation Learning for Trajectories: Recently, deep learning has been used for spatial-temporal data mining [6, 18, 39, 42] such as next location prediction [6, 19] and trajectory embedding [7, 17, 39]. Yao et al. [39] used a recurrent network with manually features to cluster trajectory into several clusters. Gao et al. [7] proposed to identify users in one mobility dataset via a recurrent network encoder. However, the uniqueness of an individual in one dataset does not imply that this user can be easily recognized in another dataset. Besides, this method can not be applied to new users whose data has not been trained with the model. Based on one dense taxi trajectory dataset, Li et al. [17] used a seq2seq model to measure the similarity of sub-trajectories extracted from one dense trajectory, which requires high data quality and fails to model the relationship of trajectories from different datasets.

Due to the limitation of model design, all these existing methods are not suitable for user identity linkage problem for different mobility data. Compared with these methods, our model is not only designed to extract comprehensive trajectory features but also model the correlated relationship between trajectories. This interaction ability is based on the attention mechanism introduced by Bahdanau et al. [1] in neural machine translation task. Feng et al. [6] is the first work to introduce the attention model to predict human mobility. However, instead of achieving mobility prediction, our focus is to measure the trajectory similarity from the different data source to link online IDs.

6 CONCLUSION

In this paper, we investigated the task of user identity linkage by leveraging the power of deep learning. We proposed an end-to-end deep learning framework to link different accounts from heterogeneous mobility data. Proposed model employs location encoder and trajectory encoder to model the complicated single trajectory feature and apply co-attention based selector to focus on discriminative parts when matching two mobility trajectory. Extensive experiments on two real-life mobility datasets show that *DPLink* significantly outperforms eight baselines on the user identity linkage task. Compared with the existing solutions, the proposed model achieves general similarity measurement for heterogeneous mobility data. Besides, it is robust to the noise of trajectory and is easy to extend to external information like PoI distribution.

There are several future directions for our work. First, because of the limitation of the datasets, we only consider simple external geographical context data like PoIs category. In the future, we plan to expand the multi-modal embedding module to process the raw textual information in the check-in data. Second, although the pre-training mechanism works well, it is still not easy to directly train a good model for sparse mobility data. Thus, design more stable network structure and better training mechanism is also an important direction.

7 ACKNOWLEDGMENTS

This work was supported in part by The National Key Research and Development Program of China under grant 2017YFE0112300, the National Nature Science Foundation of China under 61861136003, 61621091 and 61673237, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

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] Wei Cao, Zhengwei Wu, Dong Wang, Jian Li, and Haishan Wu. 2016. Automatic user identification method across heterogeneous mobility data sources. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 978–989.

[3] Alket Cecaj, Marco Mamei, and Franco Zambonelli. 2016. Re-identification and information fusion between anonymized CDR and social network data. *Journal of Ambient Intelligence and Humanized Computing* 7, 1 (2016), 83–96.

[4] Wei Chen, Hongzhi Yin, Weiqing Wang, Lei Zhao, and Xiaofang Zhou. 2018. Effective and Efficient User Account Linkage Across Location Based Social Networks. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1085–1096.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] 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 (WWW)*. International World Wide Web Conferences Steering Committee, 1459–1468.

[7] 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 (IJCAI)*.

[8] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. 2013. Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22nd international conference on World Wide Web (WWW)*.

[9] Oana Goga, Patrick Loiseau, Robin Sommer, Renata Teixeira, and Krishna P Gummadi. 2015. On the reliability of profile matching across large online social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1799–1808.

[10] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779–782.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation* 9 8 (1997), 1735–80.

[12] Shouling Ji, Weiqing Li, Neil Zhenqiang Gong, Prateek Mittal, and Raheem A Beyah. 2015. On Your Social Network De-anonymizablity: Quantification and Large Scale Evaluation with Seed Knowledge.. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[13] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, and Raheem Beyah. 2014. Structural data de-anonymization: Quantification, practice, and implications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1040–1053.

[14] Ethan Katz-Bassett, John P John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. 2006. Towards IP geolocation using delay and topology measurements. In *Proceedings of the ACM SIGCOMM conference on Internet Measurement (IMC)*.

[15] Nitish Korula and Silvio Lattanzi. 2014. An efficient reconciliation algorithm for social networks. *Proceedings of the VLDB Endowment* 7, 5 (2014), 377–388.

[16] Shamanth Kumar, Reza Zafarani, and Huan Liu. 2011. Understanding User Migration Patterns in Social Media. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[17] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S Jensen, and Wei Wei. 2018. Deep Representation Learning for Trajectory Similarity Computation. (2018).

[18] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. 2019. DeepSTN+: Context-aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. In *AAAI*.

[19] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[20] Chris YT Ma, David KY Yau, Nung Kwan Yip, and Nageswara SV Rao. 2013. Privacy vulnerability of published anonymous mobility traces. *IEEE/ACM Transactions on Networking (TON)* (2013).

[21] Nehal Magdy, Mahmoud A. Sakr, Tamer Mostafa, and Khaled El-Bahnasy. 2016. Review on trajectory similarity measures. In *IEEE Seventh International Conference on Intelligent Computing and Information Systems*.

[22] Farid M Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, and Martin Vetterli. 2016. Where you are is who you are: User identification by matching statistics. *IEEE Transactions on Information Forensics and Security (TIFS)* (2016).

[23] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*.

[24] Christopher Riederer, Yunsung Kim, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. 2016. Linking users across domains with location data: Theory and validation. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*. 707–719.

[25] Luca Rossi and Mirco Musolesi. 2014. It's the way you check-in: identifying users in location-based social networks. In *Proceedings of the second ACM Conference on Online Social Networks (COSN)*.

[26] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. 2011. Quantifying location privacy. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*.

[27] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018–1021.

[28] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.

[29] Mudhakar Srivatsa and Mike Hicks. 2012. Deanonymizing mobility traces: Using social network as a side-channel. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 628–637.

[30] Zhen Tu, Kai Zhao, Fengli Xu, Yong Li, Li Su, and Depeng Jin. 2018. Protecting Trajectory from Semantic Attack Considering k-Anonymity, l-diversity and t-closeness. *IEEE Transactions on Network and Service Management* (2018).

[31] Gang Wang, Sarita Yardi Schoenebeck, Haitao Zheng, and Ben Y. Zhao. 2016. "Will Check-in for Badges": Understanding Bias and Misbehavior on Location-Based Social Networks. In *Proceedings of the International Conference on Web and Social Media (ICWSM)*.

[32] Huandong Wang, Chen Gao, Yong Li, Gang Wang, Depeng Jin, and Jingbo Sun. 2018. De-anonymization of Mobility Trajectories: Dissecting the Gaps between Theory and Practice. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[33] Huandong Wang, Chen Gao, Yong Li, Zhi-Li Zhang, and Depeng Jin. 2017. From Fingerprint to Footprint: Revealing Physical World Privacy Leakage by Cyberspace Cookie Logs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*. 1209–1218.

[34] Huandong Wang, Yong Li, Gang Wang, and Depeng Jin. 2018. You Are How You Move: Linking Multiple User Identities From Massive Mobility Traces. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 189–197.

[35] 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 (WWW*. 1241–1250.

[36] Fengli Xu, Guozhen Zhang, Zhilong Chen, Jiaxin Huang, Yong Li, Diyi Yang, Ben Y Zhao, and Fanchao Meng. 2018. Understanding Motivations behind Inaccurate Check-ins. *Proceedings of the ACM on Human-Computer Interaction (CSCW)* (2018).

[37] Ming Yan, Jitao Sang, Tao Mei, and Changsheng Xu. 2013. Friend transfer: cold-start friend recommendation with cross-platform transfer learning of social knowledge. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*.

[38] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. 2017. Multi-site User Behavior Modeling and Its Application in Video Recommendation. In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*.

[39] Di Yao, Chao Zhang, Zhihua Zhu, Jianhui Huang, and Jingping Bi. 2017. Trajectory clustering via deep representation learning. In *International Joint Conference on Neural Networks (IJCNN)*.

[40] Reza Zafarani and Huan Liu. 2014. Finding Friends on a New Site Using Minimum Information. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*.

[41] Jiawei Zhang, Xiangnan Kong, and Philip S. Yu. 2014. Transferring heterogeneous links across location-based social networks. In *WSDM*.

[42] Zefang Zong, Jie Feng, Kechun Liu, Hongzhi Shi, and Yong Li. 2019. DeepDPM: Dynamic Population Mapping via Deep Neural Network. In *AAAI*.