Residence and Workplace Recovery: User Privacy Risk in Mobility Data

Yuchen Qiu[1], Yuanyuan Qiao*, Shengmin Wang[2], Jie Yang
[1]Beijing University of the Post and Telecommunications
Xitucheng Road 10
Beijing, China, 100876
* yyqiao@bupt.edu.cn
[2]China Telecom Corporation Limited Cloud Computing Branch Corporation

Abstract

Mobility data has been collected through mobile devices and cellular networks used in academic research and commercial purposes for the last decade. Since releasing individual's mobility records or trajectories gives rise to privacy issues, datasets owners tend to only publish encrypted mobility data, which doesn't contains users' identification symbol like telephone number. However, we argue and prove that even publishing encrypted mobility data could lead to privacy problem, of which the critical problem is users' residence and workplace identification. We develop an attack system that is able to identify users' important locations by a semi-supervised learning model. In addition to traditional time features, our system takes the users' mobility and living patterns into consideration, which are important and affect each other. Our model demands for less ground truth labels and produces considerable improvement in learning accuracy. With large-scale factual mobile data and long-time tracking ground truth data captured from a big city, we reveal that our attack system is able to identify users' residence and workplace with accuracy about 98%, which indicates severe privacy leakage in such datasets. And we provide advice for this kind of privacy-preserving problem.

Introduction

With the rapidly increasing popularity of personal mobile devices and location-based applications, the researches about large-scale human living patterns are becoming more and more available (Lane et al. 2010) . People's movements could be sensed and easily collected by mobile phone, generating large scale of mobility data, such as Call Detail Records (CDR)(Jiang, Ferreira, and Gonzalez 2017), Global Positioning System (GPS) (Lin and Hsu 2014) tracks and OIDD. Although there is criticism of precision and bias of mobile big data, it's still one of the most comprehensive sources which help us discover the large-scale human mobility(Tang et al. 2015). Understanding human living patterns is of great meaning and importance, as it has the powerful potential to reveal people's social and living status, which provides key insights for planning and making decision for city. One of the most attractive directions is people's commuting and flowing, of which the key is to identify the hot and important locations in the trajectory, such as residence and work place. However, while the utility of deep learning is undeniable, training data also presents serious privacy issues. The data will be kept forever by companies and users have no methods to delete them. Although users may benefit from new services based on big data training, there are still potential privacy risks.

A plethora of important locations identification methods has been proposed, such as traditional methods(Krumm 2007), semantic analysis (Zheng et al. 2009) and machine learning(Krumm and Rouhana 2013). Some of above researches typically use strict rules to determine important locations, which may cause the lost of other meaningful locations. For example, Cao et al.(Cao et al. 2019) and Kung et al.(Kung, Sobolevsky, and Ratti 2013) imposed a filtering recognizing places which cost more than 50% of the observed dwell-times as the work/home location. But there are fatal problems why meaningful locations with 49.9% time cost are dropped and how could we retrieve them. So we are preferring more intelligent methods like deep learning which doesn't rely on strict rules. Besides, the lack of ground truth of residence and workplace identification has long been the problem to the public and researchers, especially for uncertain locations that don't follow strict rules. Liao (Liao, Fox, and Kautz 2005) let the subjects manually label all types of locations and activities with great human resource costs. As a result, with the availability of massive real trajectory data that may cover millions of people, semi-supervised and unsupervised learning methods demanding for less labels or no label are more practical for researchers.

There has been a lot of researches identifying important locations by using features of users. Concentrating on the user's performance on each location in the trajectory, Krumm(Krumm and Rouhana 2013) extracts users' time features from the data of American Time Use Survey (ATUS), such as arrival time of locations, duration of visit, visit midpoint time of day. In order to find out the classification of locations by online user flowing, Falcone(Falcone et al. 2014) put forward a set of machine learning features based on a Tweet dataset, which contains the number of visitors, Tweets entropy, frequency Tweet entropy, etc. In fact, different kinds of people act quite differently in daily life, which means identification only considering of performance in locations is not comprehensive and inaccurate. The personal representing characteristics of users should be considered together with their performance in locations.

To solve the two above challenges, we propose a semi-supervised learning Trajectory features Refinement module for LSTM network (TR-LSTM) and a model based attack system to recover important locations from encrypted data. As the traditional methods identify locations by human formulated rules, extremely limited information like a percentage is extracted from users' trajectory. In order to classify users with similar living patterns into the same class, our system pro-processes the raw data by city area gridding and topic classification. Our system extracts features from not only locations in trajectory but also users' living patterns and mobility, considering the affect of users' patterns to their performance on locations. We label the locations following the strict rules like Cao et al.(Cao et al. 2019) automatically and train our TR-LSTM model by these labeled data. As we find the trained model also has a good performance on identifying locations that miss the strict rules lightly such as our TR-LSTM, it helps us require less to ground truth data.

In this paper, we propose a deep learning based semi-supervised system to identify important locations in one week trajectory. First, users' trajectories are extracted from encrypted OIDD. To reduce the noise, extracted trajectories are put into a pre-processing layer, using an algorithm called Stable, Oscillation, Leap periods (SOL) (Qi et al. 2016) to discover and reduce oscillations. Next, we separately extract living patterns feature, mobility feature and time features of each locations in the trajectory by clustering users' moving patterns class, calculating users' radius of gyration and users' performance on locations. After that, we label locations to get a completed dataset for model training, test and verification. Finally, we apply different deep learning model like MT-DNNs, LSTM with attention and some other models (Ma et al. 2018) on the dataset to compare their identification abilities. The output layer which contains the trained model identifies users' important locations from trajectory. Based on our attack system, we give some advice on privacy protect of big data in operaters. Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to identify important locations in trajectory by a deep learning based semi-supervised method. Compared to traditional techniques, such as clustering, and machine learning methods(Falcone et al. 2014), our framework performs better with large-scale data and demands for less manual labels. In our TR-LSTM, a fixed layer was added into the forget gate to improve the ability of attention on specific features.
- We propose a trajectory processing attack system named T2vec to represent the trajectory of a person as vectors. As Krumm(Krumm and Rouhana 2013) only extracts users' time features, our system takes users' living patterns feature and mobility feature into consideration, for their affect to users' performance on locations (the time features).
- We compared identification result of our attack system labeled by our Label Layer with the ground-truth label dataset provided by operators and our system show good performance. Based on our experiments, we give some advice on privacy protect of big data in operators.

The rest of the paper is structured as follows. Section 2 reviews related work. Section 3 identifies the problem and discusses the key challenges. Section 4 proposed the framework of our system. We evaluate our proposed system with real massive data in Section 5 and provide concluding remarks in Section 6.

RELATED WORK

Important location identification methods could be divided into unsupervised learning methods and supervised learning methods by whether using labels. We summarize the related works from two aspects: unsupervised learning methods and supervised learning methods.

Unsupervised learning: Unsupervised learning methods are preferred when labeled data is rare. Ashbrook (Ashbrook and Starner 2003) proposed a unsupervised Markov model to determine different meaningful locations by clustering locations of trajectory using K-means, which firstly applied Markov in locations identification. Based on it, Liao (Liao, Fox, and Kautz 2005) put forward the improved relational Markov networks based on that with a better performance on recognizing associated places. Place Lab (Kang et al. 2004) is also a traditional unsupervised learning method to identify locations, using both K-Means and Gaussian mixture model (GMM) approach. Summarying different kinds of unsupervised learning methods, Krumm (Krumm 2007) put forward location tracks methods consisting of Last Destination, Weighted Median, Largest Cluster and Best Time, which is widely recognized by researchers. The Best Time, which identifies locations by users' time cost, is recognized as an effective and simple method by researchers until now, such as Tian(Tian, Winter, and Wang 2019) and Wei (Wei et al. 2018). Both traditional methods and Markov model have a simple structure but a normal accuracy. Latter, Semantic analysis was first applied to infer users' travel experiences and the relative interest of a location by Zheng (Zheng et al. 2009), and X Cao (Cao, Cong, and Jensen 2010) improved the model considering both location significance and user authority. But there is still a limit of semantic analysis in multi-features extracting. Our improved deep-learning based method has a more complex structure which can mine the huge amounts of information in large-scale data and different kinds of features.

Supervised learning: When labels are adequate, supervised learning methods has a better precision. With the labeled data, Krumm created the Placer (Krumm and Rouhana 2013), an principled algorithm for labeling places based on machine learning, to infer semantic places labels. The Placer attempts to automatically label places based on how an individual uses them and the surrounding businesses. Supported by a database of categories and coordinate associations (namely a Foursquare database), Falcone (Falcone et al. 2014) applied up to 6 classification algorithm containing J48, Decision Table, Multilayered Perceptron, Bayesian Network, K* and LogitBoost to find out where people eat, drink, work and study. While with better precision, the strict requirement to labels of supervised learning is daunting to researchers.

GENERAL FRAMEWORK

In this section, we give an overview of our attack system as shown in Figure 1. The framework is an unsupervised approach with four layers, detailed as followed.

The first layer (Pre-Processing Layer) is aimed to process the raw data to remove the noise. We use the second layer (Features Extracting Layer) to extract different features considering of users' mobility, living patterns and performance on locations. The third layer (Label Layer) labels locations data automatically by their time cost build a reliable dataset to train our TR-LSTM model. The forth layer (Identification Layer) finally uses the trained TR-LSTM model to identify locations as residences, work place or others.

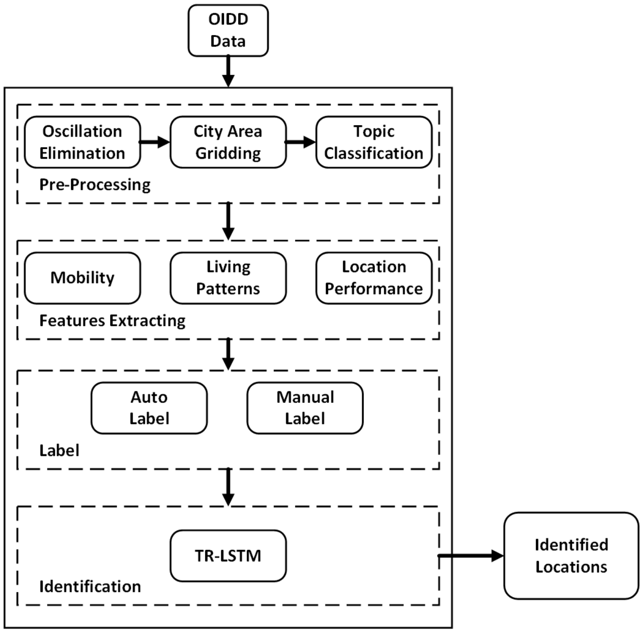

Figure 1: Overview of T2vec system

- `Pre-Processing Layer`: The input of this layer is the OIDD records sequences of each user. In this layer, we remove the invalid and low-quality data records and apply a improved SOL algorithm to reduce the noise. Before extracting the features, we divide the whole city area into 50X50 grids, of which we use the number to replace users' longitude and latitude location. In order to classify users with similar living patterns into the same class, we classify grids into different topic (e.g. suburban industrial area and suburban residential areas) by a semantic analysis method applied in our previous work, merging the in-and-out population flow and find out the symbolic topic of each user.
- `Features Extraction Layer`: In addition to users' performance on locations (time features) in the research of Krumm(Krumm and Rouhana 2013), we extract features from different aspects such as living patterns and mobility. First, we divide the whole day into four parts, which begins at 0 o'clock i.e., each part lasts 6 hours. Getting four symbol topics of each user like $(8, 7, 7, 8)$ (where the number 7 refers to the office building area and number 8 refers to the living quarters), we cluster these topic vectors into different classes as a feature representing users' living patterns. Second, we calculate the radius of gyration for each user as another feature which represents users' mobility. Third, we extract some features considering of users' performance (e.g. total time cost and frequency in the location) on each location in the trajectory. In this layer, we get a feature vector consisting of the above features for user in each location.
- `Label Layer`: The intention of Label Layer is to label locations data as residences, workplaces or others in order to train our TR-LSTM model. Based on the features vector of each locations in trajectory from last layer, Label Layer builds dataset by labeling automatically, where the trained data is totally consist of locations following the strict rules and test data contains different types of locations which may miss the rules.
- `Identification Layer`: Finally, we use the Identification Layer with trained model to identify important locations, which contains a decider to select the most convincible important locations like home and work place in a user's trajectory.

METHODOLOGY

In this section, we elaborate the layers which are key in our framework.

Pre-Processing

Consider a set of a user's raw trajectory data $O = \{o_1, o_2, ..., o_L\}$. And each stay o_l is defined as $o = (U, L, T)$, where U is the user ID; L is a two-dimensional vector (longitude and latitude) representing the users' location; T is the timestamp. We assume the user still stay at the former location until she arrives the next one. Arranging them by timestamp, we get the duration by

$$D_{former} = T_{later} - T_{former}. \quad (1)$$

Then we get a new set of $O = \{o_1, o_2, ..., o_L\}$ where o_l is defined as $o = (U, L, T, D)$. Oscillations happen when a device, even when not moving, does not only connect to the nearest cell tower, but is instead unpredictably switching between multiple cell towers because of random noise, load balancing, or simply dynamic changes in signal strength (Qi et al. 2016) . Before we extract features, the data still need oscillations processing such as SOL to detect and remove the oscillations, which recognizes the suspicious trajectory and oscillation locations by divided the oscillation into 5 Heuristic and 3 period (Ma et al. 2016) . For example, all the invalid records lacking of necessary fields and high-speed moving

Table 1: Topic classification and relevant function

Topic	Relevant function
$T1$	only appear in the afternoon and night in suburban
$T2$	development zones(centered at Pudong)
$T3$	urban residential areas
$T4$	suburban workspace
$T5$	science and education area
$T6$	business and entertainment area
$T7$	suburban industrial area(centered at Minxing)
$T8$	suburban residential areas

(faster than the local traffic speed limit) records between two closely cell towers will be removed.

Preparing for classifying users with similar living patterns into the same class, we divide the whole city where the record marks exist into 50X50 grids. By merging the in-and-out user flow using a topic-based inference model put forward by Yuan (Yuan, Zheng, and Xie 2012), we classify grids into different class. We regard a city grid as a document, a function of grids as a topic, a movement of users as a word and the total flow of users as the frequency of a word . A typical dynamic topic model can find different topic distributions among documents through different time segments.

Features Extracting

Now that we have cleaned our data and prepared for the user classification, we aim to identify locations with multi-dimension features of users. To extract the features of locations better, we want to take more characteristics of users into consideration, like their mobility, living patterns, rather than only its performance on the locations. The extracted features may help us answer the following questions: (1) *how the user live daily?* (2) *how far is the scope of user moving?* (3) *how is the users' performance on locations?* The features identified are listed below.

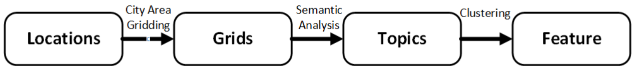

Figure 2: Procedure of living patterns feature finding

Living patterns. To find out users' the living patterns, we follow the trail from users' hottest location to hottest grid, to symbolic topics and then to living patterns classification as shown in Figure 2. First, we divide the whole day into four part, which begins at 0 o'clock and lasts 6 hours as Figure 3. And we select grids where user stay for longest time to represent the user's symbolic grid at each time part. We get the users' symbol grids as $G_u = (g_1, g_2, g_3, g_4)$ where the g is the grid ID. As we have find different topic for every grid in 4 time parts in one day listed in Table 1 in our former research, we replace the symbol grids of users to the symbol topic t_n that matches as $T_u = (t_1, t_2, t_3, t_4)$.

As each user has been clustered into different topics, we want to further discover the inner relationships between different topics in 4 time parts, such as a typical white collar is more likely to stay in a place with topic of office building in time part 2 and part 3. Therefore, we try to cluster them into groups. Besides, an integer of cluster result is more friendly to our model compared with a four-dimension vector. The most well-known hierarchical algorithms are single-link and complete-link; the most popular and the simplest partitional algorithm is K-means (Jain 2008). Considering the number of the symbol topic t_n, we try to apply K-means with different K to the T_u to minimized the error to find the best K and every user will be classified into a class K_m, which represents users' living patterns.

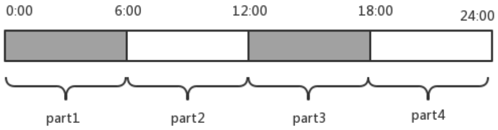

Figure 3: Overview of Time Part Divided

Mobility. We want to discover the approximate scope of users' daily moving. We calculate the radius of gyration for each user, which is simple but reflects users' mobility effectively. The radius of gyration is defined as the standard deviation of distances between the user's locations and the user's center of mass, which measures both how frequently and how far a user moves. A low radius of gyration indicates a user who travels locally, while a high one indicates a user travel between far away locations. The radius of gyration of a user can be calculated by

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - r_{cm})^2}, \qquad (2)$$

where n is the number of locations in a user's trajectory, and $(r_i - r_{cm})$ is the distance between each location and the user's center of mass r_{cm} .

Performance in locations. We extract performance features for every location in a user's trajectory by $F_n = (f_1, f_2, ..., f_{11})$ defined as Table 2 shows.

Finally, we combine the three parts above to the completed features of locations in the trajectory as $F_n = (K_m, r_g, f_1, f_2, ..., f_{11})$, composed of not only performance features of locations but also users' living patterns and mobility, which helps our model understand these locations better. There is a gap between the amounts of three aspects features, in turn which demands for a reasonable balance in model's weight.

Label

One of the problem in important locations determination with deep learning is the ground truth, since it's uneasy to

Table 2: Features of Locations Performance Definition

Feature	Definition
$f1$	total time user cost in the location,
$f2$	total day-time (from 8:00 a.m. to 8:00 p.m)
$f3$	total night-time (from 8:00 p.m. to 8:00 a.m)
$f4$	percent day-time (calculated by $f2/f1$)
$f5$	percent night-time (calculated by $f3/f1$)
$f6$	the hottest hour of the position in the day
$f7$	total frequency of the position
$f8$	total day-frequency (from 8:00 a.m. to 8:00 p.m)
$f9$	total night-frequency(from 8:00 p.m. to 8:00 a.m)
$f10$	percent day-frequency (calculated by $f8/f7$)
$f11$	percent night-frequency (calculated by $f9/f7$)

get the real residences and workplaces from users. To solve the problem, we try to use as less as possible ground truth label.

Following the rules imposed in researches of Cao et al.(Cao et al. 2019) and Kung et al.(Kung, Sobolevsky, and Ratti 2013) , we recognize those locations in line with the strict rules as important locations. Based on it, we define a strict important location as follows.

Definition 1. Each user to spend more than 50% of the total observed daytime/nighttime dwell-times for the place to be identified automatically as the strict work/home location.

As there are some possible locations missing the 50% lightly, we define a light important location as follows.

Definition 2. Each user to spend less than 50% but more than 40% of the total observed daytime/nighttime dwell-times for the place to be identified manually as the light work/home location or others.

All the strict location data will be labeled automatically as residences or workplace. A light location fitting the percent rules may be a real residence, workplace or just other locations should have been cleaned. We get their label from the ground truth dataset. The manually labeled light locations data will be used to verify our experiments as the validation set but not to train the model.

In this layer, we get a completed dataset, of which the training set is consist of 100% automatically labeled strict important locations data and the test set and verification set both have 30% labeled light important locations data. It makes our semi-supervised learning model better understand different kinds of locations with less labels from ground truth dataset.

Identification

In this layer, we first train the deep learning model such as MT-DNNs(Liu et al. 2019), LSTMs and so on embed in the system by our new build dataset from the Dataset Layer. Considering the characteristics of determination of important locations and the features we extract, we propose a Trajectory features refinement module for LSTM network (TR-LSTM). As the F_n contains 13 features $(K_m, r_g, f_1, f_2, ..., f_{11})$, there are 11 performance features more than the living patterns one and mobility one. As a result, the weights will possibly amass in the 11 features if without a limit, which may invalidate the former two features K_m and r_g.

There has been LSTM with attention (Wang et al. 2016) which could concentrate on given features. Attention is a mechanism combined in the LSTM allowing it to focus on certain parts of the input sequence when predicting a certain part of the output sequence, enabling easier learning and of higher quality. Combination of attention mechanisms enabled improved performance in many tasks making it an integral part of modern LSTM networks. But the auto-focus and weight reducing mechanism of LSTM with attention may not perform well on our dataset because of our datasets' less features, which may contribute to attention to other 11 features and decreasing of all weights.

The forget gate in LSTM controls the extent to which a value remains in the cell. The output of forget gate is calculated by

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{3}$$

The value of f_t is between 0 and 1 accordingly to amount of info we want to remember. There is no doubt that we want the forget gate keep informations of the former two features K_m and r_g as much as possible. We modify the forget gate by adding a fixed module after the σ, which will forget more about the redundancy informations of the latter 11 features. Another weight matrix W_{fixed} was embed into the module in order to raise the output of former two features as follows

$$f_t = W_{fixed} \cdot (\sigma(W_f \cdot [h_{t-1}, x_t] + b_f)). \tag{4}$$

As is shown in Figure 4, the output of specified cells will be fixed to a higher value rather than an output of sigmoid based on the h_{t-1} and x_t. Next, the whole trajectory of a user is converted into a multidimensional vector, consisting of the features vectors of each locations as the second features layers does. Then the vector will be send into a decision device to find out its identification by the three-dimensional result vector that the trained model predicts. Finally, the decider outputs the result of important locations identification in the user's trajectory.

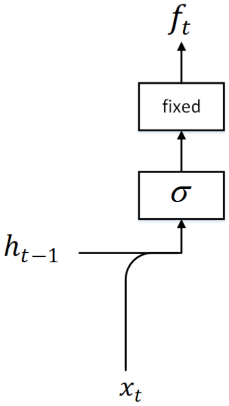

Figure 4: Fixed Module Forget Gate in TR-LSTM

EXPERIMETNS

In this section, our methodology is implemented and applied on our real origin datasets. First, we provide a brief overview of our origin dataset. Then the dataset is put into our four-layer system and the deep learning model in the system will be trained. Finally, we further apply different deep learning model in the system to our new build dataset and compare the value of prediction accuracy.

Origin Dataset

The origin OIDD comes from a large Chinese 2G/3G/4G service provider which contains approximately 3 million users in Shanghai from Jan 14, 2019 to Jan 28, 2019. Data was collected by wireless towers every time users make phone calls, send messages or change their locations. Because of the precision of towers, there are mean errors below 100 meters. The collected data comprises a sequence of records ordered by timestamp, containing anonymous identity of mobile device, visiting time, record type and location like $(6953...S39U, 20190118101357, 3, 121.69259, 31.39048)$. For privacy reason, all the personal information has been irreversibly encrypted, and this process doesn't affect the result of our analysis.

Another ground truth dataset was collected by long-time tracking by operators, of which make use informations far more than that of one week records of our experiments. There are encrypted identity, residence location, workplace location and their confidence like$(6953...S39U, Location1, Location2, 99\%, 99\%)$. Some residence and work place data was calculated with an 99% confidence as operators claim, which are target of our attack experiments.

Former Layer

The former three layers help us build the new dataset from origin real dataset, which is related to some parameters referenced to the origin data.

First, we use the SOL to solve the oscillation of origin data. As the SOL has divided the records into 3 stable period heuristics, T_n and D_n are the time and distance threshold for each stable period heuristic and V is the max speed threshold. With reference to the SOL applied to GSM and TD-SCDMA networks (Qi et al. 2016), we set the parameters with $T1 = 20s, T2 = 10s, T3 = 56s, D1 = 6.0km, D2 = 0.6km, D3 = 10km, V = 105m/s$, for the reason of that the coverage of CDMA cell station is about twice that of GSM. After that, by using the typical dynamic topic model, we classify the function of grids in 4 time parts into 8 classes, which means every grid may function in one of the 4 time parts as one of 8 types, as is shown in Table 3 .

Table 3: Grids Function Classification Results Examples

Grid	Window1	Window2	Window3	Window4
15	6	6	7	8
24	3	6	7	6
72	3	7	7	4
2279	6	6	3	6

Next, we apply K-Means to the grids vectors with K from 2 to 40 , and get the result as Figure 5. As it shows, we get the better performance when k=23 considering a decline between the loss when k=22 and k=24. Then we cluster users into 23 classes as a feature which represent living patterns.

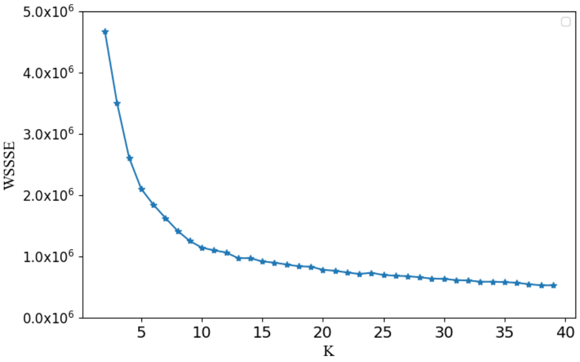

Figure 5: K-Means K and WSSE

Then the former three layers help us build a totally new dataset based on the origin real data, which will be used to train our deep learning model in the identification layer. We implemented the framework with Python and Keras. All the experiments were performed on a server with Intel Xeon CPU and Nvidia GTX 1080Ti.

Results

We compare our method with three residences and workplace identification methods based on different measures, including random method, traditional rules, MT-DNNs. Besides, we compare above methods which contains deep learning model with different dataset with full and half features.

The multi-class classification problem refers to assigning each of the observations into one of k classes. As two-class problems are much easier to solve, many authors prefer to use two-class classifiers for multi-class classification. We measure our identification results in precision, recall, accuracy, F1 micro-average, and F1 macro-average [11], defined as follows.

True Positive (TP) stands for the number of locations that match. Finally, we measure the accuracy of each method, computed as follows:

$$Accuracy = \frac{TP}{Trajectories} \tag{5}$$

The macro-average is the mean of all the labels' F1-scores, thus attributing equal weights to each F1-score. And the micro-average is calculated by the summation of contingency matrices for all binary classifiers with equal weights, which means to divide one 3-classes classifier into three 2-classes classifier, so that F1 scores of larger classes affect the metric more than smaller classes.

$$macroP = \frac{1}{n}\sum_{1}^{n} P_i \tag{6}$$

Table 4: Accuracy of locations identification for applying different algorithms to full dataset with full features

Method	Accuracy	MacroP	MacroR	MacroF1	MicroP	MicroR	MicroF1
Random	0.555	0.333	0.333	0.333	0.002	0.002	0.002
Traditional	0.651	0.738	0.598	0.661	0.003	0.003	0.003
MT-DNN	0.960	0.905	0.955	0.929	0.024	0.024	0.024
LSTM-Attention	0.855	0.830	0.814	0.822	0.007	0.007	0.007
TR-LSTM	**0.986**	**0.982**	**0.981**	**0.982**	**0.067**	**0.067**	**0.067**

Table 5: Accuracy of locations identification for applying different algorithms to full dataset with half features

Method	Accuracy	MacroP	MacroR	MacroF1	MicroP	MicroR	MicroF1
MT-DNN	0.895	0.812	0.881	0.845	0.009	0.009	0.009
LSTM-Attention	0.879	0.876	0.855	0.865	0.008	0.008	0.008
TR-LSTM	**0.919**	**0.894**	**0.879**	**0.886**	**0.012**	**0.012**	**0.012**

$$macroR = \frac{1}{n}\sum_{1}^{n} R_i \tag{7}$$

$$macroF1 = \frac{2 \times macroP \times macroR}{macroP + macroR} \tag{8}$$

$$microP = \frac{\overline{TP}}{\overline{TP} \times \overline{FP}} \tag{9}$$

$$microR = \frac{\overline{TP}}{\overline{TP} \times \overline{FN}} \tag{10}$$

$$microF1 = \frac{2 \times microP \times microR}{microP + microR} \tag{11}$$

For the random method, we randomly classify the locations into three different types (home, work and others). For the traditional method which couldn't find out the light locations, we can only classify those locations meeting the strict rules.

First, we apply the method above in the full dataset. The determination performance of different methods is shown in Table 4, where the Accuracy is the mean of three classifies' accuracy. As is shown in the table, our system with trained deep learning model have a better performance in accuracy than traditional method, for the reason of successful determination of light important locations. LSTM with attention performs worse than MT-DNN probably because of too much attention to the latter 11 features of users' performance on the location. For the reason, our improved model TR-LSTM performs better than other baseline methods.

Next, apart from the random and traditional method, we respectively apply our methods with different deep learning model to full and half features dataset, where the half features dataset contains only the latter 11 features of locations. The identification performance of different methods with different dataset is shown in Table 5, where all the performances are worse than those in Table 4. With no former two features, the models performs worse than before especially in identification of light important locations. It shows that living patterns and mobility features extracted in second layer helps us better determine those important locations in the trajectory. Our model embedded in the system trained by totally strict locations dataset successfully determine both strict locations and light locations that traditional methods may miss.

Advice

First, the successful attack experiments of our system shows that simple identity encrypted users' trajectory data may also reveals users' privacy information like residence and workplace, which could directly point to a specific user. Second, the better performance of full features dataset shows that the more personal information (living patterns and mobility) exposed to big data or AI system, the more specific person is.

For better privacy protect, identity like IMEI in data had better been encrypted with salt and changed at times within one day to avoid the whole day tracking and personal living features reveal.

CONCLUSION

In this paper, we use a deep learning semi-supervised model embedded in our attack system to determine important locations in users' trajectory. We propose a Trajectory features refinement module for LSTM network (TR-LSTM) and a important locations determination system to transform origin data to new build features dataset which will be used to train our model, and successfully identify important locations. The experiment shows that our semi-supervised system is capable of identifying important locations based on short-term data. In the future, we plan to build a OIDD privacy protect algorithm based on our giving advice, which will be used to help operators better protect users privacy while sharing data.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61671078, 61701031), in part by the Beijing Laboratory of Advanced Information Networks of BUPT, in part by the Beijing Key Laboratory of Network System Architecture and Convergence of BUPT, in part by the National Key Research and Development Program of

China (2017YFB0802701), in part by the Fundamental Research Funds for the Central Universities (2018XKJC04), in part by the 111 Project of China (B17007), and in part by the Data Intelligence Innovation and Entrepreneurship Platform conducted in the Cooperation between the Beijing University of the Post and Telecommunications and China Telecom Corporation Limited Cloud Computing Branch Corporation.

References

Ashbrook, D., and Starner, T. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* 7(5):275–286.

Cao, H.; Xu, F.; Sankaranarayanan, J.; Li, Y.; and Samet, H. 2019. Habit2vec: Trajectory semantic embedding for living pattern recognition in population. *IEEE Transactions on Mobile Computing* 1–1.

Cao, X.; Cong, G.; and Jensen, C. S. 2010. Mining significant semantic locations from GPS data. *PVLDB* 3(1):1009–1020.

Falcone, D.; Mascolo, C.; Comito, C.; Talia, D.; and Crowcroft, J. 2014. What is this place? inferring place categories through user patterns identification in geo-tagged tweets. In *6th International Conference on Mobile Computing, Applications and Services, MobiCASE 2014, Austin, TX, USA, November 6-7, 2014*, 10–19.

Jain, A. K. 2008. Data clustering: 50 years beyond k-means. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part I*, 3–4.

Jiang, S.; Ferreira, J.; and Gonzalez, M. C. 2017. Activity-based human mobility patterns inferred from mobile phone data: A case study of singapore. *IEEE Transactions on Big Data* 3(2):208–219.

Kang, J. H.; Welbourne, W.; Stewart, B.; and Borriello, G. 2004. Extracting places from traces of locations. In *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, WMASH 2004, Philadelphia, PA, USA, October 1, 2004*, 110–118.

Krumm, J., and Rouhana, D. 2013. Placer: semantic place labels from diary data. In *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13, Zurich, Switzerland, September 8-12, 2013*, 163–172.

Krumm, J. 2007. Inference attacks on location tracks. In *Pervasive Computing, 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007, Proceedings*, 127–143.

Kung, K. S.; Sobolevsky, S.; and Ratti, C. 2013. Exploring universal patterns in human home/work commuting from mobile phone data. *CoRR* abs/1311.2911.

Lane, N. D.; Miluzzo, E.; Lu, H.; Peebles, D.; Choudhury, T.; and Campbell, A. T. 2010. A survey of mobile phone sensing. *IEEE Communications Magazine* 48(9):140–150.

Liao, L.; Fox, D.; and Kautz, H. A. 2005. Location-based activity recognition using relational markov networks. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, 773–778.

Lin, M., and Hsu, W.-J. 2014. Review. *Pervasive and Mobile Computing* 12(Complete):1–16.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4487–4496. Florence, Italy: Association for Computational Linguistics.

Ma, Z.; Xue, J.-H.; Leijon, A.; Tan, Z.-H.; Yang, Z.; and Guo, J. 2016. Decorrelation of neutral vector variables: Theory and applications. *IEEE transactions on neural networks and learning systems* 29(1):129–143.

Ma, Z.; Lai, Y.; Kleijn, W. B.; Song, Y.-Z.; Wang, L.; and Guo, J. 2018. Variational bayesian learning for dirichlet process mixture of inverted dirichlet distributions in non-gaussian image feature modeling. *IEEE transactions on neural networks and learning systems* 30(2):449–463.

Qi, L.; Qiao, Y.; Abdesslem, F. B.; Ma, Z.; and Yang, J. 2016. Oscillation resolution for massive cell phone traffic data. In *Proceedings of the First Workshop on Mobile Data, MobiData@MobiSys 2016, Singapore, June 30, 2016*, 25–30.

Tang, J.; Liu, F.; Wang, Y.; and Wang, H. 2015. Uncovering urban human mobility from large scale taxi gps data. *Physica A: Statistical Mechanics and its Applications* 438:140 – 153.

Tian, Y.; Winter, S.; and Wang, J. 2019. Identifying residential and workplace locations from transit smart card data. *Journal of Transport and Land Use* 12(1).

Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 606–615. Austin, Texas: Association for Computational Linguistics.

Wei, Q.; She, J.; Zhang, S.; and Ma, J. 2018. Using individual gps trajectories to explore foodscape exposure: A case study in beijing metropolitan area. *International Journal of Environmental Research and Public Health* 15(3).

Yuan, J.; Zheng, Y.; and Xie, X. 2012. Discovering regions of different functions in a city using human mobility and pois. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, 186–194.

Zheng, Y.; Zhang, L.; Xie, X.; and Ma, W. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, 791–800.