

Generating Synthetic Trajectory Data Using GRU

Xinyao Liu¹, Baojiang Cui^{1,*} and Lantao Xing²

¹Beijing University of Posts and Telecommunications, Beijing, 100876, China

²Nanyang Technological University, Nanyang Avenue, 639798, Singapore

*Corresponding Author: Baojiang Cui. Email: cuibj@bupt.edu.cn

Received: 06 May 2021; Accepted: 15 June 2021

Abstract: With the rise of mobile network, user location information plays an increasingly important role in various mobile services. The analysis of mobile users' trajectories can help develop many novel services or applications, such as targeted advertising recommendations, location-based social networks, and intelligent navigation. However, privacy issues limit the sharing of such data. The release of location data resulted in disclosing users' privacy, such as home addresses, medical records, and other living habits. That promotes the development of trajectory generators, which create synthetic trajectory data by simulating moving objects. At current, there are some disadvantages in the process of generation. The prediction of the following position in the trajectory generation is very dependent on the historical location data, but the relationship between trajectory positions tends to be ignored. Most commonly used methods only adopt the probability distribution of users' positions to generate synthetic data. On the one hand, this type of statistical method is too rough, and on the other hand, it cannot bring more benefits in availability by increasing data volume. We propose a new trajectory generation method in this paper—Trajectory Generation Model with RNNs(TGMRNN), to address the deficiencies above. It adopts the RNN model to replace the traditional Markov model to generate trajectory data with higher availability. Meanwhile, it solves the problem that RNNs are unsuitable for continuous location data by representing trajectories as discretized data with the grid method. We have conducted experiments in a real data set. Compared with the Markov model, the results of TGMRNN demonstrate that it is superior to some existing methods.

Keywords: Synthesis trajectory; GRU; grid method

1 Introduction

Current intelligent services are data-driven, and the provision and optimization of service are based on massive user data. And when service providers use user' data, they will inevitably violate the user's privacy. With the development of communication technology, positioning technology has been more widely applied. Mobile phones, computers, cars, and other devices, used in our daily life are all equipped with positioning technology. Newzoo Global Mobile Market Report, published in 2020, shows that The number of global



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

smartphone users will reach 3.5 billion in 2020, up 6.7% year-on-year, and the number will reach 4.1 billion in 2023. Research on location-based services has made significant progress. Three main factors promote its rapid development. The first factor is that people are aware of the importance of human mobility research, which is very important to guide social construction, such as disease transmission, urban planning, and personal location services. The second factor is the rapid development of sensor devices. Our ability to collect location data information has been significantly improved with the rapid popularization and development of sensor devices (such as smartphones, wearable devices, onboard sensors, etc.). The third factor is the emergence of premium location services. More and more location-based services and applications, such as AutoNavi, Didi, Meituan, and others, collect users' locations to understand their spatiotemporal mobility patterns. And they are based on location to provide users with convenient life and rich entertainment [1]. But, some malicious applications and unreliable third parties, which are mounted on these devices and provide a variety of location-based services (LBS), may violate users' location privacy by acquiring and analyzing the subscriber's location information [2,3]. The lawbreakers can infer the user's sensitive information, such as a family address, medical records, religion, and other privacy. They may commit a crime based on location privacy. Therefore, a reliable and effective location privacy protection mechanism is urgently needed by users.

Trajectory generators can well solve the above problems. Namely, on the premise of guaranteeing the users' location privacy, it can publish the highly available synthetic trajectory data [4–6]. The trajectory generators extract features and model the moving pattern of users' to train the model. The synthetic data generated by the trained model should be highly similar to the original data, and it also could be used in data mining. At present, the main research content in trajectory generation is how to improve the availability of generated data under the premise of privacy. Researchers invest in the analysis and mining of mobile objects' location information and produce many research results. But most of the researches focus on analyzing the historical trajectory of moving objects and mining exciting information, and there are a few types of research on trajectory generation technology.

At present, the mainstream trajectory generation method is to generate synthetic trajectory data through the statistical distribution of the data set, constructing prefix tree or the Markov model etc. Compared with the availability, researchers pay more attention to protecting the privacy of the generated data. Hence, more research on the availability of the generated data needs to be invested [7,8]. The Markov model used in the current study has some shortcomings. First, the Markov model is not enough to process some complex data, and the accuracy of the Markov model is relatively low. Second, the simple low-order Markov model works better than the high-order Markov model in the location prediction. Still, low-order Markov can not obtain the context of the high dimensional sequence [9].

To address these problems, we have presented the Trajectory Generation Model with RNNs (TGM-RNN), which analyzes the historical location data of moving objects to generate synthetic trajectory data. Recursive neural networks can learn and model these hidden patterns contained in the original trajectory dataset. Trajectory data is sequential data having long-term temporal dependence. In practice, we need to analyze the trajectory data for complex and hidden movement patterns. The position sequence is sequential, and there is a context relation between the positions. The model should be able to continuously learn the characteristics of the position before and after the current position. This paper is to generate the high availability data set by processing and analyzing the historical tracks. Recurrent neural networks (RNN) can store memory through connections with feedback nerves. Besides, RNN networks have a built-in advantage in learning and generating data with long-term time dependence. Therefore, we choose to use RNNs to achieve the generation of forged data sets.

The contributions in this paper are summarized as follow:

(1) We transform location data to trajectory sequence by grid method. The location information mainly consisted of latitude and longitude, which is continuous. The grid method divides the geographic space into different cells and then takes the corresponding cell number as the number of points falling into the corresponding cell. The transformation from continuous data to discrete data is realized. And we convert the problem of continuous location generation to that of sequence generation.

(2) We adopt the GRU model to generate the trajectory, ensuring that the trajectory generator could process high-dimensional sequences and preserve the relationship between the positions. Markov model is only effective in processing low-order (1-order or 2-order) sequences relative to the GRU model. We design the training and generating process using the many-to-one pattern, combined with the N-gram method, which can customize the length of the processing sequence.

(3) We conduct experiments on a real dataset, demonstrating the effectiveness of TGMRNN compared with the mainstream model.

The rest of this paper is organized as follows: we review relevant related work in Section 2. Section 3 describes the Trajectory Generation Model using RNNs in detail. The experimental results and performance analysis are presented in Section 4. Section 5 concludes this paper.

2 Related Work

This section describes related work on trajectory data generation based on Markov models and RNN models.

Mechanisms for generating trajectory data need to model the sequential data. Markov model is a standard sequence generation method. It models the past sequence behaviour of users to generate the next behaviour of users. Some studies use the Markov model to predict the next place for trajectory generation. Simmons et al. [10] trans the Hidden Markov Model (HMM) for each user to indicate which place the user will. Liao et al. [11] train a hierarchical Markov model on subscribers' daily trajectories. The methods above train the model and predict the future location based on a single object's historical trajectory. In this case, when the user reaches an area that has not been reached, the model will not predict. Xue et al. [12] propose that the collective-pattern-based method divides group tracks into sub trajectories to overcome the shortcomings of modelling individuals. These sub trajectories are combined into growth tracks, and then the Markov model is used to predict the location. However, there are also drawbacks to the collective-pattern-based method. The collective-pattern-based approach is too coarse-grained in predicting the next area, such as using a first-order Markov model that predicts the same next place for all users in the same location. In some estimation generation work based on privacy protection, Xue et al. [13] conduct a prefix-tree using individual tracks and then construct a Markov model to predict the users' next locations. Chen et al. [14] utilize a hybrid-granularity prefix tree structure, which is efficient data-dependent yet differentially private, to generate trajectories. Chen et al. [15] proposed a novel approach, which extracts the essential information of a sequential database in terms of a set of variable-length n-grams with large counts to generate and release trajectory data. He et al. [16] took the prefix tree structure and a hierarchical reference system to model users' moving pattern at different velocities. Wang et al. [17] proposed a private trajectories calibration and publication system (PTCP), which generates synthetic trajectories by the build noise-enhanced prefix tree. Xu et al. [18] proposed DP-LTOP, which divides the original trajectory sequence into different sub-segments, and then selects the appropriate locations and segments to form a synthetic trajectory. Wang et al. [19] proposed DP-PSP, which addresses the heterogeneity of trajectories by the anchor point clustering and road segment mechanism to better the synthetic trajectories' usability. Gursoy et al. [20] proposed Adatrace, which consist of four steps, features extraction, feature extraction, synopsis learning, privacy and utility preserving noise injection, and generation of differentially private synthetic location traces and generates

synthetic trajectory by the Markov model with noise. This paper points out three threats, Bayesian inference threat, partial sniffing threat, and outlier leakage threat which are important but many studies have overlooked. Gursoy et al. [21] proposed DP-star, similar work to Adatrace. DP-star takes the Minimum Description Length metric to normalize the trajectory data to benefit the model training process. Ou et al. [22] adopted to extract reliable segments from sub-trajectories, build an exploration tree, and generate synthetic trajectories. Ghane et al. [23] proposed TGM, which could generate trajectories with arbitrary length, could encode trajectory data into the graphical generative model effectively.

Recurrent Neural Networks (RNNs) has been not only successfully applied in Nature Language Processing but also achieve outstanding performance for sequential data, such as text generation, image captioning [24], and location prediction [25]. RNN iteratively reads the sequence, iterating through each element of the sequence and updating its representation based on the input and the previous state. The connection between the hidden units and their respective projections is preserved. Gating units are often used in RNN models to transform the information flow in a more structured manner. A critical factor in determining the applicability of RNN is the size of the data set because RNN has poor generalization performance over small data volumes [26]. Based on the application of the current research results of RNN and the characteristics of the sequence generation problem, we adopt the RNN model to achieve trajectory generation. RNNs network is practised to process sequence data with long-term space-time dependencies, and it is ideal for achieving the inherent attributes of the continuous position.

3 TGMRNN Overview and Core Components

This section presents the architecture of TGMRNN.

3.1 Preliminaries

RNN is suitable for processing time-sequential data, and it can transfer the output and state of the current moment to the next moment as input. Therefore, this kind of string structure can maintain the relationship between moments. And there are problems of gradient disappearance and gradient explosion. Hence it is difficult for RNN to maintain long-term dependence. Researchers further create many excellent evolution models based on RNN, such as Long Short-Term Memory (LSTM) and GRU, to solve these problems. These models solve long-term dependence by adding memory units and avoiding gradient explosions by gating units. Compared with LSTM, GRU has fewer parameters and faster training.

The proliferation of digital mobile data, such as GPS tracking, wireless communication records, and social media location records, coupled with the superior predictive power of artificial intelligence, has sparked rapid development in the field of human mobility research. Trajectory generation technology is an essential part of human mobility research. We want to take feature extraction from a data set containing mobile users' actual location trajectories to build a generation model and achieve the purpose of maintaining statistical utility. The research on human mobility mainly includes three aspects: the next location prediction, the people flow forecast, and trajectory generation. Mobile communication networks, GPS, and social networks are the primary source of location data. For example, the mobile communication network has developed into the fifth generation of mobile communication technology. The mobile communication network can almost cover all the range of human activities by various heterogeneous networks (such as satellite networks). The mobile communication devices will interact with the base stations when we use mobile phones and other mobile communication devices to send and receive information. In this interaction process, the users' geographical location will be acquired by the mobile communication network.

Mobility data describes the movement of a group of individuals during the observation period, usually collected by sensors, and is stored as a spatiotemporal trajectory or flow of movement. An individual

trajectory is a group of records, typically each containing the identification, the geographic location, and the timestamp of the location.

The format of a trajectory is defined as follows:

Definition 1 (Trajectory). Let i denote a unit and a trajectory of i is $T_i = [ST_1, ST_2 \dots ST_{n_i}]$, which is composed of n_i sequential locations visited by the unit i . A spatiotemporal location, denoted by ST , contains the geographic location and the timestamp, and $ST = (l, t)$. Let $l = (\textit{latitude}, \textit{longitude})$ because the geographic location of a unit is usually represented by latitude and longitude. And t represents the timestamp of the corresponding location.

We need to discrete the continuous two-dimensional(latitude and longitude) spatial data by embedding technology in trajectory generation. The embedding technology is to divide the two-dimensional space into a limited number of independent regions and then map the location points to labels. The grid method is defined as follows:

Definition 2 (Grid method). For the geographic region A , a grid method G contains n independent regions, represented as $G = \sum_{i=1}^n g_i$, and it satisfies $\forall i \in n, g_i \neq \emptyset$ and $\forall i, j \in n, i \neq j, g_i \cap g_j = \emptyset$. And each region G has its own label. Therefore G could map geographic locations to labels.

Our goal is to extract and learn the mobility patterns of users to generate synthetic data.

3.2 TGMRNN Model

Consider a dataset of accurate location trajectories, denoted by D_{raw} . We want to design a generative model for learning and modelling the original location data and generating synthetic trajectory data set, denoted by D_{syn} . The synthetic trajectories D_{syn} need a high degree of similarity to D_{raw} .

We design and develop TGMRNN, a trajectory dataset generator. Fig. 1 illustrates the system architecture of TGMRNN.

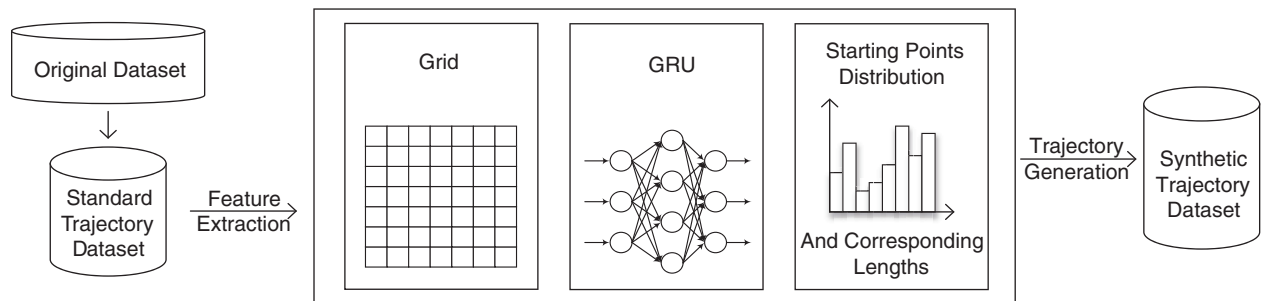


Figure 1: TGMRNN system architecture

3.2.1 Data Pre-Processing

TGMRNN preprocesses the input data set D_{raw} . Our expected trajectory data should be collected by the same sampling rate and vehicle from users, and track locations are distributed within the specified geographic area. Although there is no unified format for the publicly available trajectory data sets, they generally contain user identification, time, and geographical location, which is sufficient for training. Therefore, before we start training the model, we need to preprocess the input data set D_{raw} and transform it into our expected format, the standard format. In this process, we take to use the Grid method to transform the data, and its details are processing process is as follows:

A. Transform the input data into a location sequence. Let P_i denote a piece of data in the data set D_{raw} and $D_{raw} = [P_1, P_2 \dots P_n]$. The information P_i contains geographic location and time, and some also include

user identification, transportation, and other information. We require that the user's location should be collected at the same sampling rate. Therefore the time interval between adjacent locations in a trajectory is the same. We divide the data set D into multiple trajectories. If the time interval between two adjacent locations is greater than the time interval corresponding to the sampling rate, the front and back should be divided into two trajectories from the current location. D_{raw} is transformed to D'_{raw} by this way. There are trajectories in D'_{raw} . Let T_i denote a trajectory, and T_i consists of the sequence of locations, which could be represented as $T_i = [(lat_1, lon_1), (lat_2, lon_2) \dots (lat_n, lon_n)]$, where (lat_1, lon_1) is the latitude and longitude of the location. D'_{raw} consists of multiple trajectories, which is represented as $D'_{raw} = [T_1, T_2 \dots T_n]$. D'_{raw} preserves the locations of the moving object and the context between the locations in this way.

B. Map the geographic location to the grid cell identity. TGMRNN discretizes continuous two-dimensional (latitude and longitude) spatial data by the grid method. As shown in Fig. 2, for example, there is a trajectory T_1 in the area and $T_1 = [L_1, L_2, L_3, L_4, L_5, L_6]$ where L_i stands for a location of the trajectory. Then, TGMRNN maps the locations of the trajectories to the cell identifications of the grid to which the locations belong to. TGMRNN transforms the location sequence T_1 to cell identification sequence S_1 , where $S_1 = [C_1, C_4, C_5, C_8, C_8, C_9]$. C_i is short for *Cell_4* in Fig. 2. TGMRNN transform D'_{raw} to $D_{trajectory}$ by mapping L_i to S_i one by one, and we denote the format $D_{trajectory}$ as the standard format.

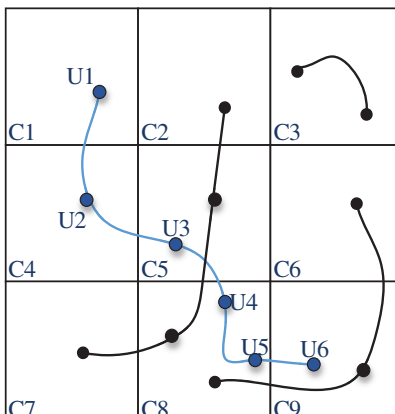


Figure 2: Grid construction. Divide maps into different levels of granularity on the different scales of the grid method

3.2.2 Training

As shown in Fig. 3, we used the RNN model to train and learn the Sequence data. The system's core function is the RNN model. The TGMRNN configures the network model to learn the convolution sequence and extend it to the space-time domain for trajectory generation. This paper adopts the GRU model. Compared with LSTM, which can remember the information in the past and selectively forget some unimportant tips to model the long-term context and other relations, GRU reduces the gradient disappearance problem while retaining the long-term sequence information. TGMRNN divides the trajectory sequence dataset into sample sequences. For each input sequence, the corresponding output contains the same length of the sequence but moves one character to the right. For example, if the sample sequence is $[C_1, C_2, C_3, C_4, C_5]$, the input is $[C_1, C_2, C_3, C_4]$, and the output is $[C_2, C_3, C_4, C_5]$. The problem can then be thought of as a standard classification problem: given the previous RNN state and the input of this timestamp, predict the class of the following sequence unit. We then added the optimizer and loss function, defined the appropriate epoch, and trained.

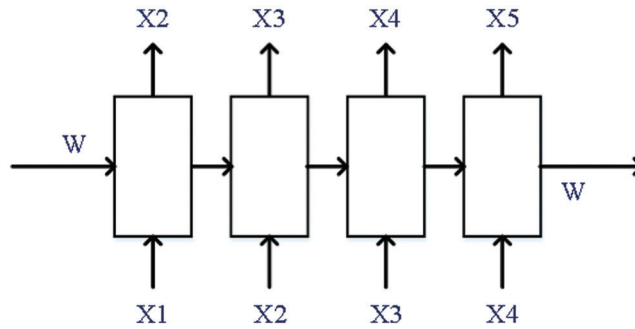


Figure 3: The process of model training

3.2.3 Generation Algorithm

Fig. 4 illustrates the basic cyclic network generation architecture. The process of sequence generation using the trained RNN model is as follows:

(1) Set the start position, initialize the RNN state, and set the trajectory's length to be generated. The starting position and RNN state are used to obtain the predicted distribution of the next position.

(2) The classification distribution is used to calculate the predicted location index, which is then used as the next input to the model.

(3) The returned state is fed back to the model. The model has more context to learn from than just one location. After the next location is predicted, the changed RNN state is fed back into the model. The model learns by always getting more context from the previously indicated place and continuously generates new sequences until the termination item is triggered.

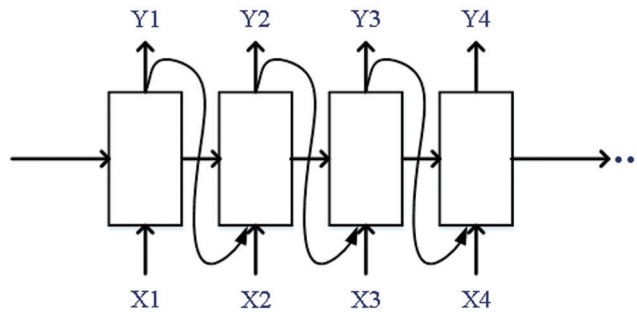


Figure 4: The process of trajectory sequence generation

4 Experiment

In this section, we report experiments conducted to assess the effectiveness of our generation algorithm.

4.1 Settings and Data Set

We implemented the experiment on Colaboratory of Google, which provided GPU for TGMRNN. We built TGMRNN by PyTorch 1.8.1 and Python 3.7.10. We compare it with the Markov model. We choose the T-Drive trajectory data sample [27], coming from Microsoft T-Drive project. It contains trajectory data of more than 10,000 taxis in Beijing for a week in 2008, which contains 15 million coordinate points and tracks over a total distance of more than 9 million kilometres. There is enough data to allow us to set up experiments of different sizes for a comprehensive comparison. We preprocessed the data set to form

tracks. After eliminating the abnormal data such as vacant position, single-point position, and error information, more than 1.8 million tracks with a length greater than one were generated in total. We compared the 1-order Markov model with the TGMRNN model during the deployment experiment because research shows that low-order Markov and high-order Markov performance is similar. To keep the training time reasonable, we set 100 epochs to train the model.

We set different numbers of mobile users, other grid partition measurements as a control experiment. We used different numbers of moving objects, 10, 100, and 1000 respectively, for training, and the corresponding number of tracks are 3215, 20306, and 172419, respectively.

We set the Markov model and TGMRNN model to learn data, respectively, generate trajectory data as the same amount of the training data, and then carry out statistical analysis.

4.2 Evaluation Metrics

We adopt Kullback–Leibler divergence (KL-Divergence) to evaluate the generated trajectory dataset's effectiveness and the real user movement trajectory dataset. KL-divergence is equivalent to the difference of Shannon entropy between two probability distributions. KL-divergence could evaluate the degree of similarity among different data sets.

Supposed $P(x)$ and $Q(x)$ are two probability distributions on the random variable x . In the case of discrete random variables, the definition of relative entropy is :

$$KL(P \parallel Q) = \sum P(x) \log \frac{P(x)}{Q(x)} \quad (1)$$

The KL-divergence is nonnegative: $KL(P \parallel Q) \geq 0$, equal to 0 when $P = Q$. The closer a KL-Divergence value is to zero, the closer the generated trajectory data set's location distribution is to the entire data set.

4.3 Evaluation of TGMRNN

We evaluate the performance of the TGMRNN and Markov models.

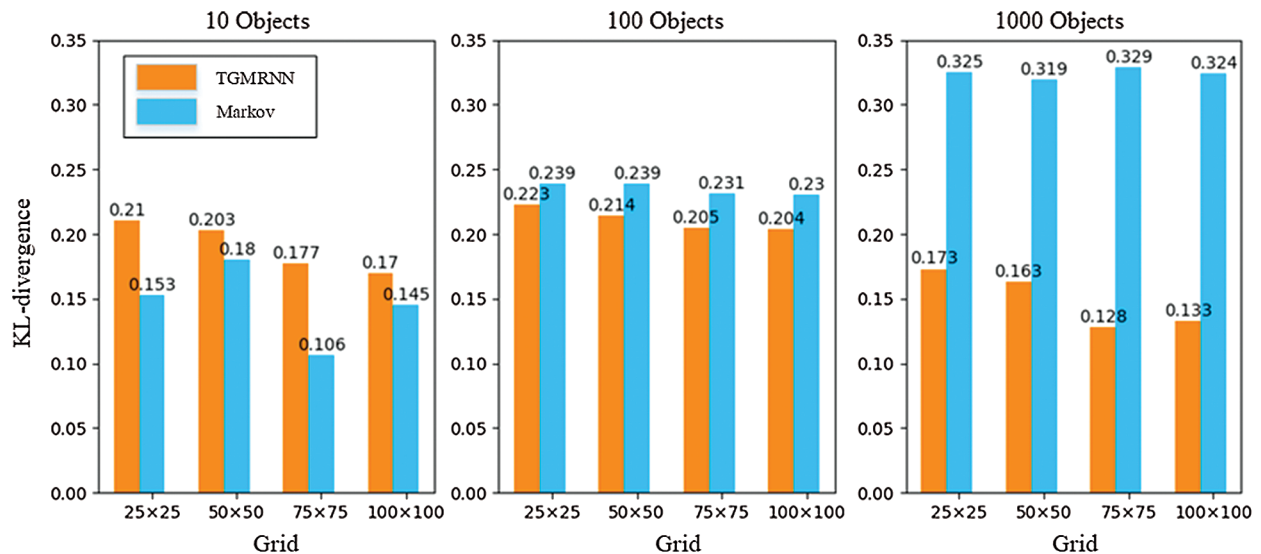
We evaluate the different influence on TGMRNN and Markov model by adopting the grid measurement 25×25 , 50×50 , 75×75 , 100×100 . The experimental results are shown in [Tabs. 1](#) and [2](#). With the increase of the value of the grid, from 25×25 to 100×100 , the map is more finely divided, and the effectiveness of synthetic trajectory data, generated by TGMRNN, gradually strengthens, where KL-divergence declines from 0.223 to 0.204 in 100 objects and from 0.173 to 0.133 in 1000 objects. It is clear that the result for 1000 objects is better than the result for 100 objects under the same grid size, and the best value occurs in 1000 objects and 75×75 . However, changes in the grid size have little impact on the results of Markov, where the values of KL-divergence stabilize around 0.233 in 100 objects and 0.324 in 1000 objects. And as shown in [Fig. 5](#), the changes in the grid size do not make a significant difference in the availability of the generated data of the Markov model. We also evaluate the effect of a change in the number of objects on the trajectory generation, with 10, 100, and 1000. In the experiment of 10 objects, the experimental results are not representative because the number of objects is too small. With the increase of the number of objects, the availability of data generated by Markov declines significantly from 100 objects to 1000 objects, whose average of KL-divergence is 0.235 and 0.324, respectively, indicating that with the rise in the number of data, the Markov model is challenging to model the trajectory data. On the contrary, the availability of TGMRNN's generated data significantly rise from 100 objects to 1000 objects. TGMRNN has better data availability than the Markov model of the same grid size and the same number of users, 100 objects and 1000 objects.

Table 1: The experiment result of the Markov model

	Grid 25×25	Grid 50×50	Grid 75×75	Grid 100×100
10 Objects	0.153	0.180	0.106	0.145
100 Objects	0.230	0.230	0.231	0.230
1000 Objects	0.325	0.319	0.329	0.324

Table 2: The experiment result of the TGMRNN

	Grid 25×25	Grid 50×50	Grid 75×75	Grid 100×100
10 Objects	0.21	0.203	0.177	0.170
100 Objects	0.223	0.214	0.205	0.204
1000 Objects	0.173	0.163	0.128	0.133

**Figure 5:** KL-divergence with different grid-scale and a different number of moving objects

5 Conclusion

We propose TGMRNN to generate synthetic trajectory data. It could produce trajectory data for mobile service providers, who need a good deal of user data to improve the quality of their products and prevent the disclosure of users' privacy. TGMRNN could transform continuous location data to discrete sequence data and preserve the relationship between the location of high-dimensional sequences. TGMRNN adopts the many-to-one pattern to train the model, which has fewer parameters and faster training and take the trained model to generate synthetic trajectory by predicting the next location. And experiments show that TGMRNN is effective.

In the future, trajectory generation is a hot research field. There are still many problems to be solved, such as (1) we should solve the geographic data sparsity to improve the availability of the trained model;

(2) we need to improve the efficiency and effectiveness of high-dimensional sequence execution; (3) we need more measures for the availability of the trajectory data; (4) we should introduce stricter privacy mechanism in trajectory generation, such differential privacy; (5) we should eliminate some privacy threats, such as Bayesian inference threat, partial sniffing threat, and outlier leakage threat.

Acknowledgement: My deepest gratitude goes first and foremost to Dr Cui, my supervisor, for his constant encouragement and guidance.

Funding Statement: The work was supported by National Natural Science Foundation of China (61941114), National Natural Science Foundation of China (Grant No. 61802025), National Natural Science Foundation of China (No. 62001055), Beijing Natural Science Foundation (4204107), Funds of “YinLing” (No. A02B01C03-201902D0).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira and L. Damas, “Predicting taxi-passenger demand using streaming data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.
- [2] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu *et al.*, “Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data,” in *Proc. 26th Int. Conf. World Wide Web*, Perth, Australia, pp. 1241–1250, 2017.
- [3] Y. A. Montjoye, C. A. Hidalgo, M. Verleysen and V. D. Blondel, “Unique in the crowd: The privacy bounds of human mobility,” *Scientific Report*, vol. 3, no. 1, pp. 193, 2013.
- [4] B. Hu and J. Wang, “Deep learning for distinguishing computer generated images and natural images: A survey,” *Journal of Information Hiding and Privacy Protection*, vol. 2, no. 2, pp. 37–47, 2020.
- [5] H. Liu, L. Cui, X. Ma and C. Wu, “Frequent itemset mining of user’s multi-attribute under local differential privacy,” *Computers, Materials & Continua*, vol. 65, no. 1, pp. 369–385, 2020.
- [6] L. Sun, C. Ge, X. Huang, Y. Wu and Y. Gao, “Differentially private real-time streaming data publication based on sliding window under exponential decay,” *Computers, Materials & Continua*, vol. 58, no. 1, pp. 61–78, 2019.
- [7] J. Yuan, Y. Zheng, C. Y. Zhang, W. L. Xie, X. Xie *et al.*, “T-drive: Driving directions based on taxi trajectories,” in *Proc. of the 18th SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, New York, NY, USA, pp. 99–108, 2010.
- [8] V. Kulkarni, N. Tagasovska, T. Vatter and B. Garbinato, “Generating Synthetic Mobility Traffic using Recurrent Neural Networks,” *ACM SIGSPATIAL Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, Redondo Beach, California, USA, pp. 1–4, 2017.
- [9] L. Song, D. Kotz, R. Jain and X. He, “Evaluating next-cell predictors with extensive Wi-Fi mobility data,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1633–1649, 2006.
- [10] R. Simmons, B. Browning, Y. Zhang and V. Sadekar, “Learning to predict driver route and destination intent,” in *IEEE Intelligent Transportation Systems Conf.*, Toronto, ON, Canada, pp. 127–132, 2006.
- [11] L. Liao, D. J. Patterson, D. Fox and H. Kautz, “Learning and inferring transportation routines,” in *Artificial Intelligence*, Amsterdam: Elsevier, vol. 171, no. 5, pp. 311–331, 2007.
- [12] A. Y. Xue, Z. Rui, Z. Yu, X. Xing and Z. Xu, “Destination prediction by sub-trajectory synthesis and privacy protection against such prediction,” in *IEEE Int. Conf. on Data Engineering*, Arlington, VA, USA, pp. 254–265, 2012.
- [13] G. Xue, Z. Li, H. Zhu and Y. Liu, “Traffic-known urban vehicular route prediction based on partial mobility patterns,” in *Int. Conf. on Parallel & Distributed Systems*, Shenzhen, China, pp. 369–375, 2010.

- [14] R. Chen, B. C. M. Fung and B. C. Desai, "Differentially private transit data publication: A case study on the montreal transportation system," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 213–221, 2012.
- [15] R. Chen, G. Acs and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proc. of the 2012 ACM Conf. on Computer and Communications Security*, New York, NY, USA, pp. 638–649, 2012.
- [16] X. He, G. Cormode and A. Machanavajjhala, "DPT: Differentially private trajectory synthesis using hierarchical reference systems," *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1154–1165, 2015.
- [17] S. Wang and R. O. Sinnott, "Protecting personal trajectories of social media users through differential privacy," *Computers & Security*, vol. 67, no. 12, pp. 142–163, 2017.
- [18] C. Xu, L. Zhu, Y. Liu, J. Guan and S. Yu, "DP-LTOD: Differential privacy latent trajectory community discovering services over location-based social networks," *IEEE Transactions on Services Computing*, vol. 1, pp. 1, 2018.
- [19] S. Wang, R. Sinnott and S. Nepal, "Privacy-protected statistics publication over social media user trajectory streams," *Future Generation Computer Systems*, vol. 87, no. 1, pp. 792–802, 2018.
- [20] M. E. Gursoy, L. Liu and S. Truex, "Utility-aware synthesis of differentially private and attack-resilient location traces," in *Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security*, New York, NY, USA, pp. 196–211, 2018.
- [21] M. E. Gursoy, L. Liu, S. Truex and L. Yu, "Differentially private and utility preserving publication of trajectory data," *IEEE Transactions on Mobile Computing*, vol. 18, no. 10, pp. 2315–2329, 2019.
- [22] L. Ou, Z. Qin, S. Liao, Y. Hong and X. Jia, "Releasing correlated trajectories: Towards high utility and optimal differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 1109–1123, 2018.
- [23] S. Ghane, L. Kulik and K. Ramamohanarao, "TGM: A generative mechanism for publishing trajectories with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2611–2621, 2019.
- [24] H. Chen, S. Li and Z. Zhang, "A differential privacy based (k- ψ)-anonymity method for trajectory data publishing," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2665–2685, 2020.
- [25] Q. Liu, S. Wu and L. Wang, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Thirtieth AAAI Conf. on Artificial Intelligence*, Phoenix, Arizona USA, pp. 194–200, 2016.
- [26] I. You, C. Choi, V. Sharma, I. Woungang and B. K. Bhargava, "Advances in security and privacy technologies for forthcoming smart systems, services, computing, and networks," *Intelligent Automation & Soft Computing*, vol. 25, no. 1, pp. 117–119, 2019.
- [27] J. Yuan, Y. Zheng and X. Xie, "Driving with knowledge from the physical world," in *Proc. of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 316–324, 2011.