Mobility Prediction at Points of Interest Using Many-to-one Recurrent Neural Network

Meng-Shiun Cheng, Jang-Ping Sheu, *Nguyen Van Cuong, and Yung Ching Kuo

*Department of Computer Science, *Institute of Communications Engineering, National Tsing Hua University, Taiwan.*

haiya25262383@gmail.com; sheujp@cs.nthu.edu.tw, cuongnv243@gmail.com, cludbuster@gmail.com

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322343

***Abstract*—With the population of mobile phones, the telecom company has a user's cellular phone signals and its movement trajectory. So, the company can learn about the user's activity habits and predict where the user may go next to meet the need of network resource and service management. In this paper, we propose a mobility prediction framework with Many-to-one Recurrent Neural Network (RNN). First, we extract the place that the user frequently visits (i.e., Points of Interest (POI)) from the user's mobility data through our proposed POI mining method, i.e., acceleration clustering. Then, we propose an adaptive mapping method to map the user's trajectory to a series of POI. Afterward, we use the RNN with Long Short-Term Memory to learn the user's POI series. Finally, we evaluate the prediction performance of the proposed scheme on two different real datasets. The performance shows that the prediction accuracy of our scheme outperforms previous works.**

***Index Terms*—Data mining, long short-term memory, mobility prediction, point of interest, recurrent neural network**

I. Introduction

BIG data prediction enables traditional heterogeneous networks to have learning capabilities and knowledge. It is an effective way to achieve network intelligence, which is a trend in modern network architecture. Proactive resource allocation (PRA) is a key technology that facilitates intelligent communication because it can take full advantage of predictions, thereby significantly improving network performance regarding throughput, energy efficiency, and quality of service (QoS). Some previous works have proved that prediction uncertainty will greatly reduce network performance [1]. Completing services for underserved users require additional resources, which results in large service latency and low throughput [2]. Mobile traffic has been increasing dramatically over the past few decades. Along with the mobility characteristic of users, multiple types of applications make the total data traffic volume of each gNB becomes more and more difficult to predict. Therefore, the mobility characteristics of user equipment have an important influencing factor on predicting gNB traffic [3].

With the widespread deployment of 3G/4G/5G cellular data networks, users access the Internet anywhere through the cellular data network with smart mobile devices, view e-mail, browse web pages, chat online, and execute a variety of mobile applications. At the same time, service and network providers have considerable potential to obtain a large amount of valuable data, particularly those related to user mobility. We can use this advantage to develop location-based services, such as location-based advertising, warning systems, and traffic planning. The points of interest (POI) represent places where the users often visit. After mapping the trajectory of a user to a series of POIs, we can select a suitable prediction method to learn the mobility pattern of the user.

POI identification and user's mobility prediction are highly meaningful in network resource and service management. First of all, it is essential in resource allocation and power-saving in the 5G network [4]. The result of POI and mobility prediction can be used to derive hotspots and crowds in a city. Then the cellular network operator can distribute more channels and network resources to offload traffic in the hotspots. Modeling user mobility at POI can improve understanding of general human movement patterns. After extracting the POI from the user's location data, we can represent the user's mobile behavior by a series of POI.

Currently, prediction methods for mobility prediction include the Hidden Markov Model (HMM) [5] and Recurrent Neural Network (RNN) [6, 7]. When using the HMM to solve the mobility prediction problem, we must make a trade-off between the order of the Markov process and the prediction accuracy. The low-order Markov process may result in low prediction accuracy because the reference movement sequence is short. However, the higher-order Markov process may suffer from the "data sparsity problem," i.e., the available historical trajectories are inadequate to cover all possible query trajectories [8]. On the other hand, earlier RNN either is tailored to specific problems or cannot be extended to long-term dependencies because the gradient of the loss function decays exponentially with time as the predictor is trained. Therefore, RNN with Long Short-Term Memory (LSTM) has been proposed as an effective and scalable model for several learning problems related to sequential data [9].

In this study, we propose a mobility prediction framework of Many-to-one RNN with Acceleration Point Mining (MRAPM) to predict the POI of mobile users. In the first step, we identify the place (POI) that the users usually visit and stay. We design a POI mining method, i.e., *acceleration clustering*, to determine the POI for each user. We obtain a user's POI from the

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant 106-2221-E-007-019-MY3, and in part by Qualcomm Technologies under Grant SOW NAT-435533.

perspective of the acceleration location of the user. In this manner, the user's POI could be identified accurately. Then, we propose the *adaptive mapping* method to map the user's movement to a series of POI. By utilizing this method, we can completely convert the user's movement trajectory into a POI series. Afterward, we design an RNN LSTM with two output layers and many-to-one architecture to learn the user's mobility pattern from the POI series. Finally, we evaluate the accuracy of the MRAPM by comparing it with other prediction frameworks and verify that the MRAPM has higher prediction accuracy than other prediction frameworks.

The remainder of this paper is organized as follows: In Section II, we review the related work. In Section III, we present the proposed MRAPM framework. In Section IV, we evaluate the performance of the MRAPM on two real datasets and compare the prediction accuracy of the proposed MRAPM framework with that of other prediction frameworks. Finally, Section V concludes this paper.

II. Related Work

In this section, we review existing works on POI identification and mobility prediction.

A. POI identification

Mining user's movement pattern by identifying POI from GPS dataset has been extensively investigated. Conventional methods used to mine the POI from the GPS dataset include the *k*-means clustering algorithm [10] and the density-based clustering algorithm [5, 6]. The *k*-means clustering algorithm is popular and effective, but the number of groups must be a known parameter. However, researchers were unable to determine the number of POI for each user. By contrast, the density-based clustering algorithm has the minimal requirements of domain knowledge and efficiency on a large dataset. The authors in [5] presented the "leader-follower clustering," which is an extension of the density-based clustering algorithm for constructing location clusters to reduce the oscillation effect and identify POI from user trajectories. The location dataset used in [5] is produced from cellular networks; thus, the position of accessed Base Stations (BSs) is used as the user position. The authors in [6, 11] proposed a two-layer method to identify the POI. They extracted the locations (i.e., stay points) where the users have stayed for a long time from the raw GPS trajectories.

B. Mobility prediction

Recognizing user mobility is an interesting topic invested in many papers. The authors in [11] summarized the algorithms and techniques used to recognize user mobility through crowd sensed data in an urban environment. Several other works use the externally acquired information, such as trip time distribution and road conditions to assist mobility prediction. Although externally acquired information is often helpful in improving prediction accuracy, the cost of obtaining externally acquired information is high. Thus, in this study, we focus on

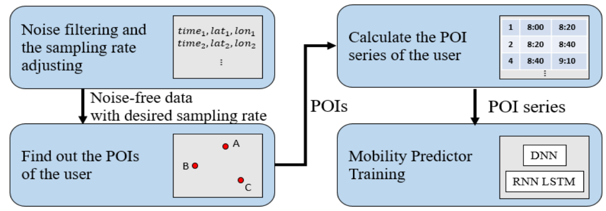

Fig. 1. Execution procedures of MRAPM.

the method that utilizes only trajectory data. In [5], the authors proposed a mobility prediction framework based on the HMM, which aims to discover individual daily time habits and predict future personal activities. However, the HMM cannot provide an in-depth reference because it is a probabilistic model, and it suffers from low prediction accuracy of trajectories with diversified and long motions. In [12], Schneider et al. analyzed networks of daily trips obtained from users of a cellular network. They applied a modified finite Markov chain embedding technique to reveal human mobility patterns, called motifs. Another work in using machine learning techniques for mobility prediction can be found in [13], where the purpose is to predict visitor distribution of users in large events. In their work, the K-means clustering algorithm is used to identify POI, and mobility prediction is derived via both classification techniques and regression techniques. To solve the previously mentioned problems, the authors in [6] adopted the RNN that can store sequential information in hidden layers and handle variable lengths of trajectories.

III. Many-to-One Recurrent Neural Network with Acceleration Point Mining (MRAPM)

The MRAPM, which includes four phases, is shown in Fig. 1. In the first phase, we preprocess a user's mobility data through noise filtering and sampling rate adjustment. In the second phase, we identify the POI of the user by using our proposed POI mining method, i.e., *acceleration clustering*. In the third phase, we map the user's movement to a POI series by using the *adaptive mapping* method. In the last phase, we use the RNN LSTM with our designed neural network architecture to learn the POI series of the user.

A. Preprocess the user's mobility data

The raw movement dataset contains some noise because of GPS signal oscillation. The noise is usually a GPS point, which is far from the user's normal movement trajectory. For example, users are moving under an underground tunnel or via a small alleyway where the GPS signal is fragile. In our work, we eliminate the abnormal GPS points that have sudden and large acceleration compared with the current acceleration. Let W denote a user's raw movement dataset. For each record W_i in W is represented as (t_i, lat_i, lon_i), which means that the user is at the location of latitude lat_i and longitude lon_i at time t_i. We take three consecutive data W_{i-1}, W_i , and W_{i+1}, and calculate the speed S_i between W_{i-1} and W_i and the speed S_{i+1} between W_i and W_{i+1} by the following formula:

$$S_i = \frac{Vincenty\left((lat_i, lon_i),(lat_{i-1}, lon_{i-1})\right)}{t_i - t_{i-1}}, \quad (1)$$

where *Vincenty* is the method used to calculate the distance between two points on the surface of a spheroid. After getting the speed, we can calculate the acceleration value A_i between S_i and S_{i+1} by the following formula:

$$A_i = \left|\frac{S_{i+1} - S_i}{t_{i+1} - t_i}\right|. \quad (2)$$

Then, we filter out W_i which has an acceleration value A_i larger than the acceleration threshold α. In our simulations, $\alpha$is set to $1.2\ m/s^2$ which is the fastest acceleration value in public transport systems. The filtered GPS dataset must have the same sampling frequency so that the density-based clustering algorithm can identify the clusters correctly. When the sampling frequency is low, it will result in insufficient data. By contrast, when the sampling frequency is high, it will cause the operation speed to slow down. In our experiments, we observe that the most suitable sampling frequency is one record every 5 s. After executing the previously mentioned step, we can generate the user's mobility dataset *D*. For each record, D_i in *D* is represented as $(t_i, lat_i, lon_i, S_i, A_i)$.

B. Identify the POI of the users

To identify the POI from the GPS dataset, we develop a POI mining method, i.e., *acceleration clustering*. In the first step of *acceleration clustering*, we select the acceleration points that start from a speed lower than 1 m/s and acceleration more than 0 m/s as the acceleration point set *A*. We observe that the place frequently visited by a user has numerous overlapping or close acceleration points. This is because when the user leaves the staying place, a significant acceleration from the low speed occurs. If it is a place frequently visited by the user, numerous acceleration points will be observed. To group the geographically similar acceleration points into a POI, we need to use the clustering algorithm in the second step.

In the second step, we apply the density-based clustering algorithm to the acceleration point set *A* to identify the POI of the user. The density-based clustering algorithm requires two parameters, namely, the radius (*eps*) and the minimum number of points required to form a cluster (*minPts*). In general, the hyper-parameters *eps* and *minPts* of the density-based clustering algorithm should be adjusted to obtain better clustering results. To automatically adjust these hyper-parameters according to the different users, we refer to the method presented in [14].

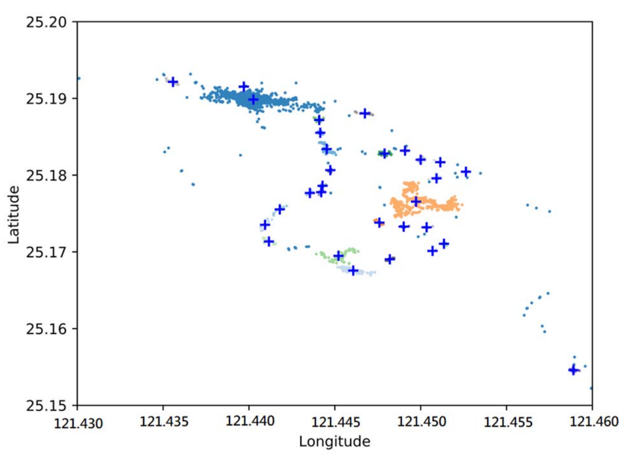

Fig. 2. POI of a user.

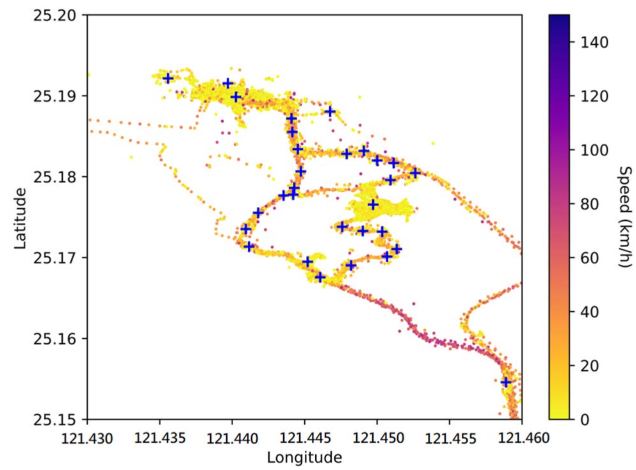

Fig. 3. Speed scatter graph with POI of a user.

We cluster the nearby acceleration points to a group and eliminate the outliers, as shown in Fig. 2. Each color in the figure represents a group, and the "+" marker denotes the geographical centroid (POI) of the group. After mapping the POI location to a real-world map, 86% of the POI are buildings. This finding indicates that we have correctly identified where the user stays frequently. Then, we place the POI marker on the speed scatter graph, as shown in Fig. 3. We observe that the user's moving speed near the POI is slower than the speed on the route between POI.

C. Calculate the POI series of the users

Here, we propose the *adaptive mapping* method to map the user's movement to a POI series. We scan dataset *D*, identify the POI area in *P* that a user has reached, and calculate the stay time of the user in the POI area. First, we need to determine the size of the POI area. If the POI area is small, then the accuracy of mobility prediction is high. However, the error range of GPS is approximately 5-10 m. Thus, the radius of the POI must be larger than 10 m. Also, if a user moves into an indoor place, we cannot detect their locations accurately. According to the results of our experiments, 50 m can best cover the user's staying range. Therefore, the POI area is set to a region within the radius of 50 m around the POI position. However, the user may not come to any POI area because the user is moving. At this time, we will set the value in the POI series to -1. After the calculation is completed, the POI series O of each user will be generated. Each element O_n in O is composed of $(\mathrm{POI}_n, sw_n, st_n, ew_n, et_n)$, where POI_n denotes the POI *n* that a user stayed in, sw_n denotes the staying start time on a certain day of the week, st_n denotes the staying start time of that day

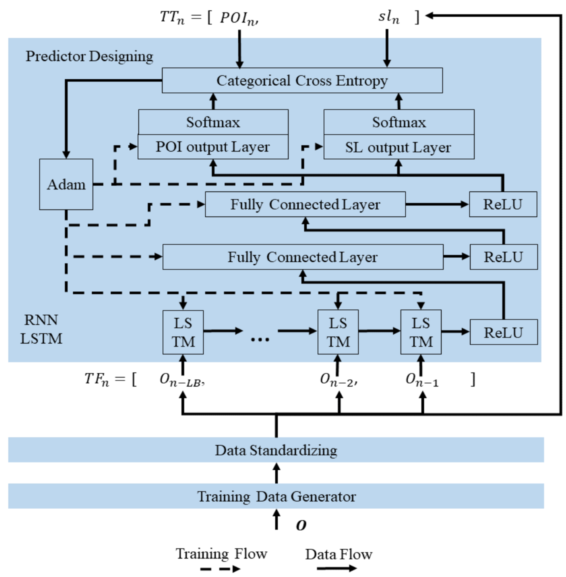

Fig. 4. Training procedure of the predictor.

in minutes, ew_n denotes the staying end time on a certain day of the week, and et_n denotes the staying end time of that day in minutes. The number of POI depends on the characteristics of the user. Normal office workers with more stable behavior patterns usually have 2 or 3 POI (home, company, dinner restaurants) on weekdays. A user with high entropy values, e.g., a salesperson may have over 20 POI because he often visits many places in a working day, as shown in Fig. 2. In general, the distribution and the number of POI will be determined by the richness of the information we have about surveying subjects. Users with a similar series of POI may take the same trajectory to achieve if their data are quite similar. However, normally, the trajectory may be different due to the different orders of POI in the process of movement and the difference of the stay time at POI.

D. *Predictor training*

In this subsection, we input the POI series O into the predictor to learn the mobility pattern of the user. As shown in Fig. 4, the predictor has three phases. In the first phase, we implement the training data generator to generate the training data from the POI series O. In the second phase, we conduct data standardization to normalize the range of all features in the training data. In the third phase, we implement the many-to-one RNN LSTM with two output layers and input the training data for training. In Fig. 4, the data flow represents the training data processing procedure, and the training flow represents the training procedure of the weights between neurons.

1) *Training Data Generator*

In this study, we use the many-to-one RNN LSTM to learn the user's mobility pattern and predict the user's next POI. First, we adjust the data format of the POI series O to meet the training format of the many-to-one RNN LSTM. Here, we define LB as the looking back number, which is the number of records that will be used to predict the next time slot. Each training data TD_n contains the training features TF_n and the training targets TT_n. TF_n can be expressed as $TF_n = [O_{n-LB}, O_{n-LB+1}, \dots, O_{n-2}, O_{n-1}]$, where n is the current record number. The training target TT_n is the tuple of the current POI and the stay time level. TT_n can be expressed as $TT_n = [(POI_n, sl_n)]$, where sl_n is the stay time level. To determine the stay time level, first, we calculate the stay time interval si_n by using sw_n, st_n, ew_n, and et_n as defined in section II-C. Then, we use the following rule to derive the stay time level:

$$sl_n = \begin{cases} \left\lfloor \frac{si_n}{10} \right\rfloor & \text{if } \frac{si_n}{10} < 6 \\ 6 & \text{else.} \end{cases} \tag{3}$$

The stay time level is an integer between 0 and 6. Stay time level 0 means the user's stay time is 0–9 min, stay time level 1 means the user's stay time is 10–19 min, and so on. Stay time level 6 means that the user's stay time is greater than or equal to 60 min.

2) *Data Standardizing*

Standardizing the training data could accelerate the training process and reduce the probability of falling into a local optimum. Moreover, weight decay and Bayesian estimation can be easily accomplished using standardized inputs [15]. Thus, we use the min-max scaling to standardize the training data. The min-max scaling is done via the following equation: $x^{std} = (x - x_{min})/(x_{max} - x_{min})$.

3) *Predictor Designing*

The many-to-one RNN LSTM is achieved by connecting several LSTM units and letting only the last LSTM unit output the computation value of the neurons. Then, we need to add an activation function so that the output signal does not become a simple linear function. Here, we use the Rectified Linear Unit (ReLU) as the activation function. The ReLU can reduce the computation cost and make neural networks sparse to alleviate the overfitting problems. Then, we add two fully connected layers to increase the learning depth of the predictor. The number of neurons is gradually reduced from the first layer to the second layer. Each fully connected layer would be activated through the ReLU before yielding the output. Finally, we add the POI output layer to output the prediction of the next POI and the stay-level output layer to output the stay time level in the next POI. We select the Adaptive moment estimation (Adam) [16] as the optimizer to adjust the weights between neurons in the training flow as shown in Fig. 4.

IV. Performance Evaluation

In this section, we compare the prediction accuracy of the MRAPM with that of other mobility prediction frameworks, including the HMM [5], grid space RNN, and LSTM-PPM [6].

A. *Experiment Definition*

We calculate the prediction accuracy by using the equation *NH/NT*. *NH* indicates the number of GPS points correctly predicted (i.e., in the correct POI area) in the predicted time interval. *NT* indicates the number of total GPS points of a user in the predicted time interval. Next, we calculate the user's entropy by using the method proposed in [5] to quantify the user's living habits. We divide the time of day into 48-time slots. Then, each user will use vector $E = [e_1, e_2, \dots, e_{48}]$ to represent

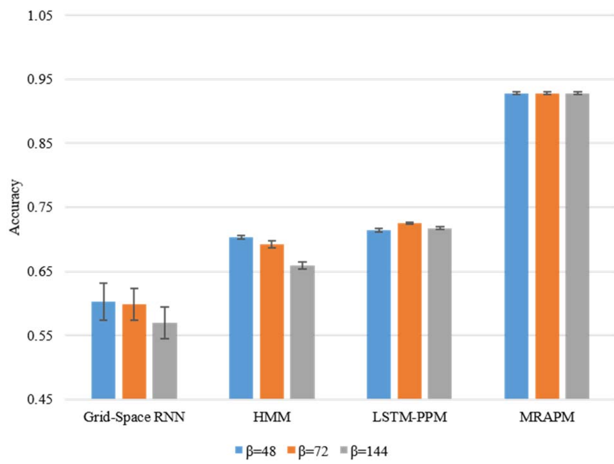

Fig. 5. The prediction accuracy of MRAPM compared with other prediction frameworks.

a day's entropy distribution. Each element e_i in E means the entropy of the time slot i. The formula for calculating entropy is

$$e_i = -\sum_{j=1}^{|P|} p_i(j) \ln p_i(j), \qquad (4)$$

where $p_i(j)$ is the probability of the user staying at POI j in time slot i and $|P|$ denotes the total number of the user's POI.

B. Environment

We selected 40 users from real-world datasets, in which 20 users are from the SensingGO [17], and 20 users are from the GeoLife dataset [18]. For each user, we assign 90% of the data for identifying the POI and training the predictor. Meanwhile, the remaining 10% of the data is assigned for evaluating the prediction accuracy. The dataset provided by the SensingGO is generated by installing a mobile software on the experimenter's mobile phone to obtain all information, such as temperature, light, and GPS, which can be collected by the mobile phone, and periodically send it back to the host for data storage. The GeoLife dataset has been used in [6, 18] and many other studies related to user movement prediction to evaluate the performance of the proposed algorithm. We use the MATLAB HMM kit to implement the mobility prediction frameworks based on HMM. The MRAPM is implemented using the Python Keras suite for the RNN LSTM architecture and the scikit-learn DBSCAN suite for density-based clustering.

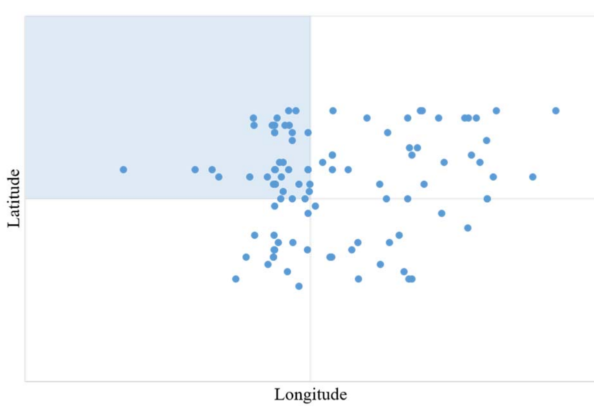

Fig. 6. The predicted area coverage of the Grid-Space RNN.

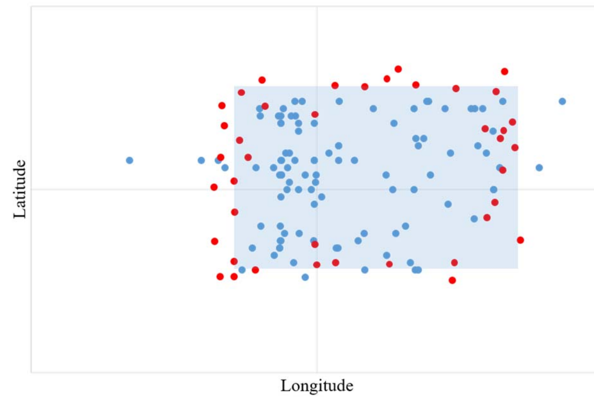

Fig. 7. The predicted area coverage of the MRAPM.

C. Results

First, we analyze the effect of different back-reference numbers LB on the accuracy of the MRAPM. We average the prediction accuracy of 40 users for different back-reference numbers LB with a 95% confidence interval. We observe that the accuracy of the MRAPM does not increase with the LB but instead initially increases when LB equals 24 and subsequently decreases. Therefore, we set the LB =24. Then, we compare the prediction accuracy of the proposed MRAPM with that of other mobility prediction frameworks, i.e., HMM, LSTM-PPM, and grid space RNN. In Fig. 5, we show the prediction accuracy of the MRAPM and other prediction frameworks with different numbers of time slots (β) for one day with a 95% confidence interval. We observe that the MRAPM has better prediction accuracy than other mobility prediction frameworks. Given that the MRAPM adopts the adaptive mapping method, its prediction accuracy would not change when β varies. For the grid space RNN, HMM, and LSTM-PPM, as β increases, the interval of the time slot decreases. The small-time slot interval will make the user's movement pattern more complicated because of the small differences in stay time and make the predictor's learning effect worse. On the other hand, as the number of time slots decreases, the generated POI series cannot provide an accurate description of the user's movement. Thus, the prediction results cannot cover the user's movement well.

Given that the MRAPM adopts the learning architecture of the RNN LSTM, the correlation of the long sequential POI distribution can be learned so that the MRAPM is less affected by the user's entropy. Then, we compare the prediction accuracy of the MRAPM and grid space RNN. The blue points in Fig. 6 represent the GPS points of a specific user on a certain

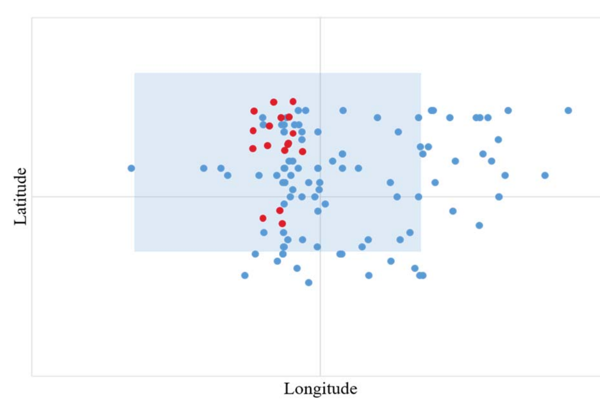

Fig. 8. The predicted area coverage of the LSTM-PPM.

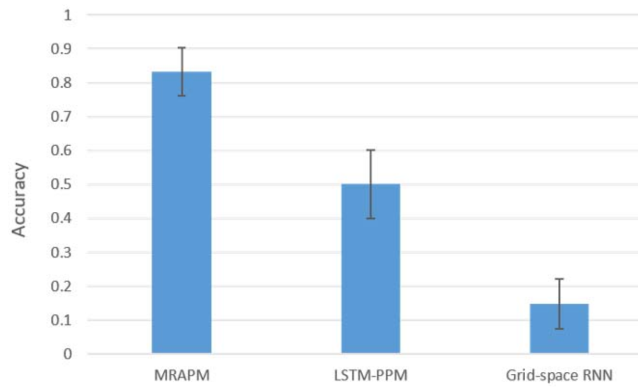

Fig. 9. The POI identification accuracy.

day between 8:00 and 8:30, and the blue rectangle represents the POI area generated by the grid space RNN framework. We observe that the POI area cannot cover the user's mobility well such that numerous GPS points are out of the POI area. In Fig. 7, the red points indicate the acceleration points of the user, and the blue rectangle indicates the POI area generated by the POI identification process of the MRAPM. We observe that the acceleration points scatter around the POI area, and the centroid of the cluster can be identified correctly. Thus, the POI area generated by the MRAPM can better cover the user's position than the POI area generated by the grid space RNN.

Furthermore, we compare the prediction accuracy of the LSTM-PPM and MRAPM. In Fig. 8, the blue rectangle represents the POI area generated by the LSTM-PPM, and the red points represent the stay points generated by the LSTM-PPM. We observe that numerous blue points cannot be covered by the POI area because it is determined using only a few stay points. In Fig. 9, with a 95% confidence interval, we show that the POI identification accuracy generated by the MRAPM is better than that generated by the LSTM-PPM and grid space RNN. This finding can be attributed to the scenario in which the POI locations generated by the grid space RNN and LSTM-PPM are biased because of the previously mentioned reasons when conducting the POI identification process. Note that HMM is a probability model for mobility prediction without POI generation capability. Therefore, it is not included in the POI identification accuracy comparison in Fig. 9.

V. Conclusion

In this paper, we propose the MRAPM, a mobility prediction framework. We identify the POI from the perspective of the acceleration points of the user and verify that the POI area generated by the *acceleration clustering* method could accurately cover the user's mobility. Then, we use the *adaptive mapping* method to map the user's trajectory to a POI series. In this manner, we could completely record the information of the user's movement. Finally, we use the POI series to train the RNN LSTM. We design the RNN LSTM with two output layers, which could predict not only the next POI that the user will go to, but also predict the stay time in the next POI. In the experiments, we evaluate the accuracy of the MRAPM on real-world datasets. We compare the accuracy of the MRAPM with that of other mobility prediction frameworks. We show that the MRAPM has better prediction performance than LSTM-PPM and Grid-Space RNN by 33% and 68%, respectively.

References

[1] J. Li, X. Zhang, J. Zhang, J. Wu, Q. Sun, and Y. Xie, "Deep Reinforcement Learning-Based Mobility-Aware Robust Proactive Resource Allocation in Heterogeneous Networks," IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 1, pp. 408-421, Nov. 2019.

[2] J. Li, X. Zhang, S. Wang, and W. Yi, "Proactive resource scheduling with time and frequency domain coordination in heterogeneous networks," *in Proc.* of *IEEE Pers. Indoor Mobile Radio Communications* (PIMRC), Bologna, Italy, Sep. 2018.

[3] S. Wang, F. Li, H. Ni, L. Xu, M. Jing, J. Yu, and X. Wang, "Rush Hour Capacity Enhancement in 5G Network Based on Hot Spot Floating Prediction," *IEEE International Conferences on Ubiquitous Computing & Communications* (IUCC), Shenyang, China, Oct. 2019.

[4] H. Farooq, A. Asghar, and A. Imran, "Mobility prediction empowered proactive energy saving framework for 5G ultra-dense HetNets," *in Proc. of IEEE Global Communications Conference* (GLOBECOM), Abu Dhabi, United Arab Emirates, Dec. 2018.

[5] Q. Lv, Y. Qiao, N. Ansari, J. Liu, and J. Yang, "Big data-driven hidden Markov model based individual mobility prediction at points of interest," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5204-5216, Sep. 2016.

[6] M. H. Wong, V. S. Tseng, J. C. C. Tseng, S. W. Liu, and C. H. Tsai, "Long-term user location prediction using deep learning and periodic pattern mining," *in Proc. of International Conference on Advanced Data Mining and Applications* (ADMA), pp. 582-594, Singapore, Nov. 2017.

[7] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," *in Proc. of AAAI Conference on Artificial Intelligence*, pp. 194-200, Phoenix, Arizona, USA, Feb. 2016.

[8] A. Y. Xue, J. Qi, X. Xie, R. Zhang, J. Huang, and Y. Li, "Solving the data sparsity problem in destination prediction," *International Journal on Very Large Data Bases*, vol. 24, no. 2, pp. 219-243, April 2015.

[9] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, July 2016.

[10] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, "A tale of ten cities: Characterizing signatures of mobile traffic in urban areas," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2682-2696, Dec. 2016.

[11] Y. Zhou, B. P. L. Lau, C. Yuen, B. Tuncer, and E. Wilhelm, "Understanding urban human mobility through crowdsensed data," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 52–59, Nov. 2018.

[12] C. Schneider, V. Belik, T. Couronné, Z. Smoreda, and M. González, "Unravelling daily human mobility motifs," *Journal of the Royal Society Interface*, vol. 10, no. 84, p. 20130246, July 2013.

[13] J. Violos, S. Pelekis, A. Berdelis, S. Tsanakas, K. Tserpes, and T. Varvarigou, "Predicting visitor distribution for large events in smart cities," *in Proc. of IEEE International Conference on Big Data and Smart Computing* (BigComp), Kyoto, Japan, Feb. 2019.

[14] M. T. H. Elbatta, and W. M. Ashour, "A dynamic method for discovering density varied clusters," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 1, pp. 123-134, Feb. 2013.

[15] S. A. Yaseen, O. Q. Aziz, and B. H. A. Bakar, "Prediction of shear strength of ultra high-performance reinforced concrete deep beams without stirrups by neural network," *Eurasian Journal of Science & Engineering*, vol. 3, no. 1, pp.142-164, Sep. 2017.

[16] D. P. Kingma, and J. L. Ba, "Adam: A method for stochastic optimization," *in Proc. of International Conference on Learning Representations*, pp. 1-15, San Diego, California USA, May 2015.

[17] Y.-H. Lin, J.-C. Chen, C.-Y. Lin, B.-Y. Su, P.-Y. Lee, "Poster: SensingGO-toward mobile/cellular data measurement with social and rewarding activities," *in Proc. of the 24th Annual International Conference on Mobile Computing and Networking*, New Delhi, India, pp. 765 - 767, Oct. 2018.

[18] Y. Zheng, X. Xie, and W. Y. Ma, "GeoLife: A collaborative social networking service among users, location, and trajectory," *IEEE Data Engineering Bulletin. 33*, pp. 32-40, Jun 2010.