CABIN: A Novel Cooperative Attention Based Location Prediction Network Using Internal-External Trajectory Dependencies

Tangwen Qian[1,2], Fei Wang[1(✉)], Yongjun Xu[1], Yu Jiang[1,2], Tao Sun[1,2], and Yong Yu[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{qiantangwen,wangfei,xyj,jiangyu,suntao}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
yuyong171@mails.ucas.edu.cn

Abstract. Nowadays, large quantities of advanced locating sensors have been widely used, which makes it possible to deploy location-based service (LBS) enhanced by intelligent technologies. Location prediction, as one of the most fundamental technologies, aims to acquire possible location at next timestamp based on the moving pattern of current trajectories. High accuracy of location prediction could enrich and increase user experience of various LBSs and brings lots of benefits to service providers. Lots of state-of-the-art research try to model spatial-temporal trajectories based on recurrent neural networks (RNNs), yet fails to arrive at a practical usability. We observe that there exists two ways to improve through attention mechanism which performs well in computer vision and natural language processing domains. Firstly recent location prediction methods are usually equipped with single-head attention mechanism to promote accuracy, which is only able to capture limited information in a specific subspace at a specific position. Secondly, existing methods focus on external relations between spatial-temporal trajectories, but miss internal relations in each spatial-temporal trajectory. To tackle the problem of model spatial-temporal patterns of mobility, we propose a novel Cooperative Attention Based location prediction network using Internal-External trajectory dependencies correspondingly in this paper. We also design and perform experiments on two real-world check-in datasets, Foursquare data in New York and Tokyo cities. Evaluation results demonstrate that our method outperforms state-of-the-art models.

Keywords: Attention · Internal-external relations · Spatial-temporal trajectory · Location prediction

1 Introduction

Nowadays, large quantities of various sensors (e.g., GPS-devices, radar system, electronic toll collection, infrared distance meter, etc.) are deployed to track

I. Farkaš et al. (Eds.): ICANN 2020, LNCS 12397, pp. 521–532, 2020.
https://doi.org/10.1007/978-3-030-61616-8_42

persons or vehicles, which makes a variety of location data accumulate steadily. Fusing different kinds of location data for one object is the key to improve relevant technologies in complex application scenarios. Location prediction, as one of the most fundamental technologies, aims to acquire possible location at next timestamp based on the moving pattern of current trajectories. In last decades, it has already been applied broadly ranging from city management to personal services. High accuracy of location prediction is fundamental to enrich and increase user experience of various LBSs and brings lots of benefits to service providers [10,11]. Therefore, the task to design best location prediction model for various situations has attracted the attention of both academy and industry.

According to the architecture, existing location prediction methods can be roughly divided into two categories: pattern-based and model-based. Pattern-based methods [1,5,8] extract spatial-temporal patterns (e.g., sequential patterns, frequent patterns) from historical movements firstly, which are used to predict next location. Although pattern-based methods are commonly used, it's non-trivial to discover meaningful patterns which are important to the performance [12]. Therefore, model-based methods [2,3,6,7], such as Markov model and Recurrent Neural Network (RNN), are introduced to tackle this problem. These model-based methods leverage sequential statistical models to capture the transition regularities of movements. At present, RNN-based methods achieve a state-of-the-art performance.

However, trajectories are too complex in some of the real-world scenarios. In these scenarios, on one hand the sensing procedure could be sheltered or disturbed, on the other the persons or vehicles with malicious purpose might try to forge location data and avoid being tracked. Therefore, relying on only short-range patterns or transition regularities might cause the huge error in prediction and crash down the LBS systems. When we try to solve the challenge with RNN, we observe that the receptive field of RNN is weak in capturing long-range dependency due to how it models and optimizes. Hence until now, many scholars continue to improve RNN-based methods to get better results in different ways, such as, replacing basic RNN with long short-range memory (LSTM) or gated recurrent neural networks (GRU) and adopting the encoder-decoder architectures. However, the fundamental constraint of sequential computation remains.

To overcome the constraints, recently scholars equip RNN with attention mechanism to improve the ability of modeling sequence context, which is proved to perform well in computer vision and natural language processing domains. Yet we observe that there still exists two ways to improve. Firstly recent location prediction methods are usually equipped with single-head attention mechanism to promote accuracy, which is only able to capture limited information in a specific subspace at a specific position. Secondly, existing methods focus on external relations between spatial-temporal trajectories, but miss internal relations in one spatial-temporal trajectory.

Inspired by above observations, we proposed CABIN, a novel **C**ooperative **A**ttention **B**ased location prediction network using **I**nternal-**E**xternal trajectory dependencies in this paper. Firstly, we transformed raw sparse trajectory data into dense feature representation with a spatial-temporal feature disentangling module. Secondly, we constructed our method based on pure attention, which is able to seize not only external but also internal relations of each spatial-temporal trajectory. Finally, we designed a cooperative attention module to effectively filter the current trajectory features with the historical spatial-temporal mobility pattern from different representation subspaces at different positions. We conducted thorough experiments on two public check-in datasets of the real world, results showed that our method reaches a new state-of-the-art result in location prediction task.

Our main contributions are summarized as follows:

- We introduced a novel complete Transformer network through introducing a spatial-temporal feature disentangling module, which is a pure attention-based Transformer network to predict next location based on historical and current trajectories.
- We proposed a new cooperative attention module and added it to traditional Transformer network to filter current trajectories based on historical ones, which is able to capture trajectory information from different representation subspaces at different positions.
- We evaluated our methods through extensive experiments on two public check-in real-world datasets. Experimental results demonstrate that acc@1 of our method improves nearly 4.79% and 9.62% than the state-of-the-art methods on the two datasets.

The rest of the paper is organized as follows. In Sect. 2, we introduce related work of pattern-based methods, model-based methods and Attention mechanism. Then our proposed method are detailed in Sect. 3. We conduct comparative experiments and perform extensive analysis of experimental results in Sect. 4. Finally, we introduce future work and conclude our paper in Sect. 5.

2 Related Work

2.1 Pattern-Based Methods

Pattern-based methods extract patterns (e.g., sequential patterns, frequent patterns) from the law of historical movements first, and then use them to predict the next location. Cheng et al. [1] focus on personalized point-of-interest (POI) recommendation in location-based service and fuse matrix factorization with geographical and social influence. WhereNext [8] is a classical pattern-based method, building a decision tree named T-pattern Tree, which is learned from the discovered patterns. The tree is then used to acquire the best matching path to predict the next location. Periodica [5] is another one classical pattern-based method. It uses reference spot to capture the reference location, and then uses a probability model to characterize the periodic behaviors. During the process of automatic pattern discovery, manual intervention is needed to judge effectiveness, which is time-consuming and inefficient.

2.2 Model-Based Methods

Model-based methods are introduced to tackle inherent problems of pattern-based methods, and obtain a better performance than pattern-based in general. Many methods have been proposed, such as hidden Markov models (HMM) [7] and Recurrent Neural Network (RNN) based models [2,6]. Hidden Markov Model (HMM) [7] is first used to model user's historical trajectories, and then we predict the next probable location by this trained HMM model. Meanwhile, a Spatial Temporal Recurrent Neural Networks model (ST-RNN) [6] is proposed to model the spatial and temporal contexts, and achieve the state-of-the-art results in the location prediction task. Until now, RNN-based methods are the most popular. Hence many scholars continue to improve the RNN-based method to get better results in recent years, such as, DeepMove [2] replaces basic RNN with more powerful GRU and extends GRU with attention mechanisms to get a higher performance. Although RNN is designed to tackle timing problem and performs well in sequence modeling, it is still weak and time-consuming in capturing long-range dependency due to its modeling and optimization mechanism.

2.3 Attention Mechanism

Attention mechanisms induce conditional distributions over hidden feature representation to compose a weighted normalized vector for feature importance evaluation. It is widely used in many fields, for examples, image classification, recommendation system, machine translation and location prediction. Armed with attention mechanism, deep learning models obtain a boosting performance and improvement on interpretability through visualizing attention matrix. For RNN-based model, attention mechanism strengthens the ability in capturing the long-range dependencies to some extent. Following the tremendous success of attention mechanism, several variant have been proposed. Among them, self-attention is extremely powerful in modeling the inherent relation between different elements in one sequence, making it suitable to perform feature combination and pattern exploration. Building on pure self-attention, Transformer [9] is firstly proposed to address the translation tasks in Natural Language Processing (NLP). In this paper, we adapt Transformer with cooperative attention module to model the mobility patterns in trajectory data.

3 Proposed Method

As shown in Fig. 1, Our method consists of two core parts, Spatial-Temporal Feature Disentangling and Attention-based Model. In former, we introduced a spatial-temporal feature disentangling module to enable Transformer network to capture spatial and temporal information from trajectories. In latter, we equipped pure attention-based Transformer network with cooperative attention module to acquire the internal and external historical and current patterns of mobility from different representation subspaces at different positions. Due to that we trained our method in an end-to-end manner, hand-crafting features are no longer needed.

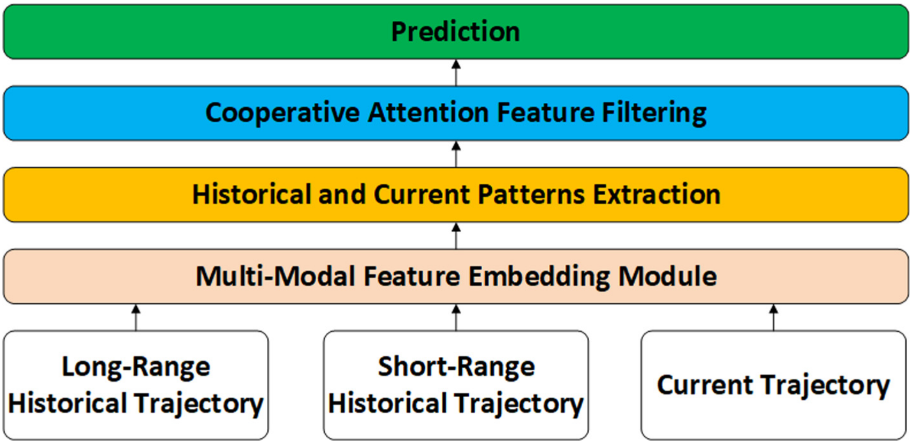

Fig. 1. Architecture of our model.

3.1 Spatial-Temporal Feature Disentangling

An embedding module is needed to transform high dimensional discrete raw features into low dimensional dense representation, which is more semantic expressing and computable. There are multiple factors that may exert influence on mobility transitions, such as exact time of day and location. It is necessary to integrate all these meaningful information together to describe target objects' movements. Therefore, we designed a spatial-temporal feature disentangling module to jointly embed the spatial-temporal features into dense representations.

Embedding Strategy. In RNN, the recurrence mechanism endows model with auto-regressive essence, making it naturally temporal perceptive. However, in pure attention-based Transformer, in order to make the model to use the order of the sequence, positional encoding is added to inject some information about the relative or absolute position of the token in the sentence. In vanilla Transformer, the positional encoding is designed as below:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{2}$$

where pos is the position and i is the dimension. d_{model} is the dimensionality of input and output. More details could be found in Transformer [9].

Considering spatial-temporal trajectories have natural time attribute which is more accurate to express the position of the input token, we replaced position encoding with temporal feature encoding in CABIN. The procedure of embedding temporal feature is as follows. Firstly, we divided temporal feature into two parts, workday and weekend. The workday is denoted as $\{0, 1, ..., 23\}$, and the weekend is denoted as $\{24, 25, ..., 47\}$. Secondly, we translated temporal feature into one-hot vectors. And finally, we mapped the high dimensional sparse one-hot vectors to low dimension dense representation.

The spatial information is the carrier of semantics in mobility of trajectories. In order to distinguish them from temporal information, we used another matrix to embed spatial information into a different semantic space.

Integration Strategy. We considered a frequently appearing operation to integrate features carrying different semantic information. As shown below:

$$x_{vanilla} = E_{spatial} + E_{temporal} \tag{3}$$

the $E_{spatial}$ and $E_{temporal}$ separately denote the dense representation after embedding of spatial and temporal features. The "add" operation assumes that these two features have same dimension. However, this is not always the case. In location prediction problem settings, input features are specific in two aspects. 1) spatial and temporal information express different meanings, which makes it inappropriate to embed them into the same hidden space. 2) The capacity of semantic space that spatial and temporal information require is vastly different, because the range of temporal information is limited in a small number of positive integers.

Simply adding these two different embeddings may confuse the model and be harmful for further feature extraction. So here we replaced the "add" operation with the "concatenate" operation. As stated below:

$$x = Concat(E_{spatial}, E_{temporal}) \tag{4}$$

3.2 Attention-Based Model

The original trajectories are divided into history ones and current ones. Seizing the patterns of mobility from trajectory data is the key to accurately predicting next location. Lots of state-of-the-art research try to equip RNN with single-head attention, which captures limited external relations between trajectories in a specific subspace at a specific position. To overcome the shortage, our method equipped pure attention-based Transformer network with cooperative attention module to seize the internal and external historical and current patterns of mobility from different representation subspaces at different positions.

Historical and Current Patterns Extraction. Transformer [9], a new network architecture, eschews recurrence and relies entirely on attention mechanism to draw global dependency between features in different positions. The most appealing strength of Transformer is that it breaks down the auto-regressive assumption to obtain the ability of highly parallel computation and one-hop feature correlation: input elements interact with each other simultaneously without regard to their distance. As a powerful model, Transformer is firstly designed to address the translation tasks in Natural Language Processing (NLP). Recently, it is proved that the architecture and capacity of Transformer makes it suitable to process massive data, such as images and videos. Considering that attention mechanism is suitable to catch internal and external spatial-temporal patterns of mobility, we designed our method based on Transformer with encoder-decoder architecture.

Each layer in Transformer Encoder module is composed of two sub-layers, multi-head self-attention mechanism (MA) and position-wise feed-forward network (FFN). And a residual connection is employed around each of the two sublayers, followed by layer normalization (LN). After N layer's feature extraction, we obtained the output O_{TE}^{N} as final historical pattern representations, here the right corner mark TE denotes the "Transformer Encoder".

Each layer in Transformer Decoder module is composed of three sub-layers, masked multi-head self-attention mechanism (MMA), multi-head self-attention mechanism (MA), and position-wise feed-forward network (FFN). Similar to Transformer encoder, a residual connection is employed around each of the two sublayers, followed by layer normalization (LN). The difference lies in the mask mechanism. The mask signal is designed to ensure that the prediction for current trajectory point depends only on previous trajectory points. After N layer's feature extraction, we obtained the output O_{TD}^{N} as final current pattern representations, here the right corner mark TD denotes the "Transformer Decoder".

Cooperative Attention Feature Filtering and Prediction. We designed a cooperative attention feature filtering module, which adapts multi-head attention, to effectively filter the current trajectory features with the historical spatial-temporal mobility pattern from different representation subspaces at different positions. Cooperative attention module regards the output of Transformer encoder, i.e. O_{TE}^{N}, as historical pattern representations, the output of Transformer decoder, i.e. O_{TD}^{N} as current movements representations.

The cooperative attention module is formulated as below:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{5}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{6}$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

where h denotes the number of parallel attention layers, We got the historical patterns filtered by current features as below:

$$O_{CA} = MultiHead(O_{TE}^{N}, O_{TD}^{N}, O_{TD}^{N}) \tag{8}$$

where the right corner mark "CA" denotes "cooperative attention".

After multi-head attention module, we obtained the probability of each POI at next time under the given historical and current trajectory as below:

$$Prob = softmax(O_{CA}) \tag{9}$$

4 Experiments

We conducted a series of experiments and compared our method, CABIN, with LSTM, DeepMove, DeepMove* (a variety of DeepMove), and CABIN* (a variety of CABIN) on two public Foursquare check-in datasets.

4.1 Dataset

In our experiments, we followed the datasets, preprocessing of datasets and data splitter setting as same as previous related work as described in [2,4]. We evaluated our model on two public Foursquare check-in datasets [13], which is collected in New York (NYC) and Tokyo (TKY) from Foursquare API for about 10 months, ranging from Apr. 2012 to Feb. 2013. Each of them contains 8 columns of data (i.e. User ID, Venue ID, Venue category ID, Venue category name, Latitude, Longitude, Time zone offset in minutes and UTC time). Here we used former 3 columns because the others carry more textual information than spatial and temporal information. In this paper, we only considered modeling trajectory data and leave this textual information to our future work.

We segmented the original trajectories into several sessions based on the time interval between two neighbor records. We chose 72 h as the default time interval threshold. Further, we filtered out the sessions with record less than 5 and users with session less than 5. In following experiments, for each user, we take the first 80% check-in data as the training set, the other 20% data as the evaluation set. The overall statistics of original and processed datasets is shown in Table 1.

Table 1. The overall statistics of datasets.

Dataset	Type	Raw	Cleaned
NYC	Users	1083	935
	Locations	38333	13962
TKY	Users	2293	2108
	Locations	61858	21395

4.2 Baselines

To evaluate the performance of our method, we compared CABIN with several representative methods for location prediction:

- LSTM [3]: Long short-term memory is an adaptive version of vanilla recurrent neural network. Equipped with gated mechanism, LSTM is more effective in modeling longer sequence. It represents a class of auto-regressive methods.
- DeepMove [2]: It's a state-of-the-art method for next location prediction. It adapts the ST-RNN with a historical attention module.

- DeepMove*: It's a variety of DeepMove, replacing single attention module with multi-head attention.
- CABIN*: It's a variety of our method, discarding cooperative attention mechanism.

4.3 Analysis

Overall Performance. The overall performance comparison on two public check-in datasets evaluated by $acc@k$, $ADE@k$ are illustrated in Table 2.

Table 2. Results for NYC and TKY dataset. The results with the best performance are marked in bold.

Dataset	Method	acc@1	acc@5	acc@10	ADE@1	ADE@5	ADE@10
NYC	LSTM	0.1557	0.3432	0.4068	4760.1629	1589.4411	1054.3837
	DeepMove	0.1839	0.3959	0.4480	3780.7436	1156.7545	768.2902
	DeepMove*	0.1958	0.3981	0.4532	3722.8963	1149.0330	765.6733
	CABIN*	0.1970	0.4092	0.4699	3630.3077	1129.0954	728.4006
	CABIN	**0.2016**	**0.4103**	**0.4764**	**3584.3926**	**1081.0376**	**674.9032**
TKY	LSTM	0.1426	0.3024	0.3624	6108.5688	2556.6697	1799.951
	DeepMove	0.1565	0.3168	0.3772	6030.0399	2157.6815	1437.6740
	DeepMove*	0.1594	0.3235	0.3836	5915.9244	2109.9256	1383.7364
	CABIN*	0.1618	0.3337	0.3956	5893.2292	2090.1591	1371.0090
	CABIN	**0.1640**	**0.3339**	**0.3982**	**5868.6996**	**2071.2962**	**1363.8166**

We can see that **CABIN*** and **CABIN** both outperforms all baselines in all evaluation metrics. Moreover, compared with the state-of-the-art **DeepMove**, our method **CABIN** gains a relative performance of 9.62% acc@1 in NYC dataset, and 4.79% acc@1 in TKY dataset. From evaluation results, we can conclude that multi-head self-attention models and cooperative attention mechanism both give obvious advantage to our method. The former succeeds in capturing external and internal mobility patterns simultaneously, while the latter is able to draw global dependency between historical and current spatial-temporal information effectively. **CABIN*** has poor results compared to **CABIN**, which suggests that there cooperative attention mechanism indeed seizes the relation between historical and current spatial-temporal trajectories.

In general baselines, **DeepMove**, as an adaption of recurrent neural network, equipped with a single historical attention module, shows a boosting performance compared with a vanilla **LSTM**, which suggests that there indeed exists historical mobility periodicity and that attention mechanism can promote the performance in seizing spatial-temporal contexts. **DeepMove*** performs better than vanilla **DeepMove** in both two datasets due to the powerful ability of multi-head attention, which captures trajectory information from different representation subspaces at different positions.

Time Consumption The time consumption comparison results are presented in Table 3. We chose two frequently-used standards to evaluate the time consumption: 1) training time spent on every epoch. 2) the number of epoch when model converges.

Table 3. Time consumption of different methods. “evaluation” denotes evaluation standards. “time” denotes training time per epoch (min). “converge” denotes the number of epoch when model converges.

Dataset	Evaluation	LSTM	DeepMove	DeepMove*	CABIN*	CABIN
NYC	Time(min)	0.583	70.921	5.058	7.794	9.277
	Converge(epoch)	19	20	22	16	17
TKY	Time(min)	1.568	216.955	16.147	22.330	28.928
	Converge(epoch)	26	29	22	16	18

It is clear that **LSTM** has advantage of high training speed among all methods, this is because it is a simple model without encoder-decoder architecture and attention mechanism. **DeepMove** uses recurrent models with time-consuming point-wise product-based attention mechanism to model long trajectory sequence, resulting in an extremely slow training process and relatively slow convergence.

Compared with **DeepMove**, **CABIN** is much time-saving mainly due to replacing point-wise product-based attention with scaled dot-product based attention. To prove the aforementioned point, we carried out comparative experiment between **DeepMove** and **DeepMove***. The only difference between **DeepMove** and **DeepMove*** is that the former uses point-wise product-based attention, while the latter replaces it with multi-head attention based on scaled dot-product attention. And the **DeepMove*** has a sharp drop in time consumption compared with **DeepMove**.

We can also see that **CABIN** is nearly two times of time consumption compared with **DeepMove***, this is because **CABIN** uses more than one module armed with attention mechanisms. To prove the aforementioned point, we carried out comparative experiment between **CABIN** and **CABIN***. We can see **CABIN*** costs less time and epochs due to that it uses no cooperative-attention module compared to **CABIN**.

Feature Disentangling Module Analysis. To validate the rationality of our spatial-temporal feature disentangling module, we conducted experiments with vanilla feature embedding, whose feature embedding adopts positional encoding and “add” integration strategies. The results are shown in Table 4. We can see that the performance of our spatial-temporal feature disentangling module is almost the same to the vanilla feature embedding in both NYC and TKY datasets. We inferred that the reason of the phenomenon is that the regular loss of location prediction models focuses only on next location, ignoring the temporal

Table 4. Results for NYC and TKY dataset. The results with the best performance are marked in bold.

Dataset	Method	acc@1	acc@5	acc@10	ADE@1	ADE@5	ADE@10
NYC	Ours vanilla	0.2012	0.4002	0.4659	3585.1205	1120.7728	728.6527
	Ours	**0.2016**	**0.4103**	**0.4764**	**3584.3926**	**1081.0376**	**674.9032**
TKY	Ours vanilla	**0.1645**	0.3319	0.3927	5869.0309	2089.3468	1393.8850
	Ours	0.1640	**0.3339**	**0.3982**	**5868.6996**	**2071.2962**	**1363.8166**

information. Due to that, the spatial-temporal trajectory can be viewed from two aspects with no difference, one is original spatial-temporal sequence, the other is ordered temporal sequence. Our spatial-temporal feature disentangling module captures the inner relation of both spatial and temporal from the aspect of original spatial-temporal trajectory, while vanilla feature embedding module captures just temporal relations from the aspect of ordered temporal sequence.

In a nutshell, our method, **CABIN**, is far more efficient than **DeepMove**. Although it is not as far as **DeepMove*** in time consumption, it costs less epochs to reach a higher accuracy. What's more, the evaluation in two real-world datasets show that **CABIN** is with good robustness.

5 Conclusion

In this paper, we focused on next location prediction problem, which is of tremendous importance for advanced location-based services. We proposed CABIN, a novel Cooperative Attention Based location prediction network using Internal-External trajectory dependencies, which enjoys two novel characteristics compared to previous methods: 1) Cooperative attention module is able to capture trajectory information from different representation subspaces at different positions, which is better and faster than single point-wise product attention. 2) Our method predicts more accurately and efficiently than existing RNN-based methods proved by experimental results on real-world datasets.

Considering that the check-in data is relatively sparse, we plan to extend the problem into other area, such as datasets of dense trajectory like T-drive taxi datasets, to improve our method robustly.

Acknowledgment. This work is partially supported by NSFC No. 61902376 and NSFC No. 61702487. This work is also financially supported by National Key Research and Development Program of China No. 2018YFC1407400.

References

1. Cheng, C., Yang, H., King, I., Lyu, M.R.: Fused matrix factorization with geographical and social influence in location-based social networks. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 22–26 July, Toronto, p. 2012. Canada, Ontario (2012)

2. Feng, J., et al.: Deepmove: predicting human mobility with attentional recurrent networks. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018, pp. 1459–1468 (2018)
3. Graves, A.: Supervised sequence labelling with recurrent neural networks. Stud. Comput. Intell. Springer, **385** (2012)
4. Hang, M., Pytlarz, I., Neville, J.: Exploring student check-in behavior for improved point-of-interest prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, 19–23 August 2018, pp. 321–330 (2018)
5. Li, Z., Ding, B., Han, J., Kays, R., Nye, P.: Mining periodic behaviors for moving objects. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 25–28 July 2010, pp. 1099–1108 (2010)
6. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 12–17 February 2016, Phoenix, Arizona, USA, pp. 194–200 (2016)
7. Mathew, W., Raposo, R., Martins, B.: Predicting future locations with hidden Markov models. In: The 2012 ACM Conference on Ubiquitous Computing, Ubicomp '12, Pittsburgh, PA, USA, 5–8 September 2012, pp. 911–918 (2012)
8. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: Wherenext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1 2009, pp. 637–646 (2009)
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp. 5998–6008 (2017)
10. Wang, F., Xu, Y., Zhang, H., Zhang, Y., Zhu, L.: 2flip: a two-factor lightweight privacy-preserving authentication scheme for VANET. IEEE Trans. Veh. Technol. **65**(2), 896–911 (2016)
11. Wang, F., Xu, Y., Zhu, L., Du, X., Guizani, M.: LAMANCO: a lightweight anonymous mutual authentication scheme for n -times computing offloading in iot. IEEE Internet Things J. **6**(3), 4462–4471 (2019)
12. Wu, R., Luo, G., Shao, J., Tian, L., Peng, C.: Location prediction on trajectory data: a review. Big Data Min. Analytics **1**(2), 108–127 (2018)
13. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Trans. Syst. Man Cybern. Syst. **45**(1), 129–142 (2015)