STF-RNN: Space Time Features-based Recurrent Neural Network for Predicting People Next Location

Abdulrahman Al-Molegi[1,2], Mohammed Jabreel[3,4] and Baraq Ghaleb[5]
[1]*Smart Health Research Group,* [3]*ITAKA Group, Universitat Rovira i Virgili, Spain*
[2]*University of Science and Technology,* [4]*Hodiedah University, Yemen*
[5]*School of Computing, Edinburgh Napier University, Edinburgh, UK*
abdulrahman.almolegi, mohammed.jabreel@urv.cat; B.Ghaleb@napier.ac.uk

***Abstract*—This paper proposes a novel model called Space Time Features-based Recurrent Neural Network (STF-RNN) for predicting people next movement based on mobility patterns obtained from GPS devices logs. Two main features are involved in model operations, namely, the space which is extracted from the collected GPS data and also the time which is extracted from the associated timestamps. The internal representation of space and time features is extracted automatically in the proposed model rather than relying on handcraft representation. This enables the model to discover the useful knowledge about people behaviour in more efficient way. Due to the ability of RNN structure to represent the sequences, it is utilized in the proposed model in order to keep track of user movement history. These tracks help the model to discover more meaningful dependencies and as consequence, enhancing the model performance. The results show that STF-RNN model provides good improvements in predicting people's next location compared with the state-of-the-art models when applied on a large real life dataset from Geo-life project.**

1. Introduction

Human behaviour is very complex and diverse. Mobility, as a component of human behaviour, is also complex, but its variability is lower and could be studied with more focused pattern-recognition approaches. In most cases, human mobility is analysed with the goal of predicting future behaviours. Mobility prediction is defined as the prediction of people's next location in the region that they constantly move in.

Owing to the widespread availability of mobile devices, such as tablet PCs, smartphones and smart watches, along with the rapid enhancement of their data collection abilities, we can collect a large amounts of people's mobility data at a very low cost. The availability of large amount of collected data together with the development of location-based applications and services, have attracted a special attention from both academy and industry towards building efficient methods for analysing and predicting user next movement or location. These methods aim to improve end-user applications, such as healthcare applications [1], recommendation systems [2], route planning, carpooling, meeting planners or location-based advertisements. They also aim to help the corresponding institutions in solving issues related to network management, healthcare, human computer interaction, socio-economic modelling for urban planning, public transportation planning, public safety assurance and etc [3], [4].

To predict people's future locations, learning techniques like Markov Model, Association rules, Bayesian Networks or Neural Networks (NN) are obvious candidates to be applied. One of the challenges that are faced by researchers while predicting people movements is how to transfer (adapt) these techniques to work with the context information of the movements. Also, building an accurate prediction model for all users is not an easy task and sometimes might be impossible because the next location prediction is a user specific problem. Even if the visited locations might overlap among different users, the trajectory of a user visiting different location is most likely unique. Thus, building one prediction model for each user might be preferable. Usually, building the model, i.e. discovering the frequent trajectories and locations, is performed off-line while the prediction itself is performed on-line.

Nowadays, predicting people's next location has been extensively studied. Markov Chain (MC) model has been presented in [5], [6] to infer users's next place by computing the transition probability from their mobility data logs. Similarly, hidden Markov model has been employed in [7], [8] to predict the trip destination taking into account the location characteristics or user activity transition as an unobservable parameter. A rule-based approach that discovers associations from movement transaction database is proposed in [9]–[11]. As another popular method, NN has been applied for location prediction problem in cellular communication network to reduce the traffic load by automatically update the location information of mobile user [12]–[16]. Recently, Recurrent Neural Networks (RNN) has been successfully utilized in many areas such as sequential click prediction [17], word embedding [18] and time series prediction [19], [20]. Some of the most recent studies leverage RNN to model people's mobility [21], [22]. The authors have reported that RNN achieves promising performance in comparison with traditional NN model.

Generally, users can be classified into three different

978-1-5090-4240-1/16/$31.00 ©2016 IEEE

types according to the predictability of their daily routine as follows: predictable users, expected users and random users [24]. Predictable users follow regular routines between their home place and workplace during their working days. For this type of users, it is easy to accurately predict their next locations. The last two types are characterized by highly complex movement and usually a larger number of visited locations. Therefore, it is difficult to accurately predict their next locations. Hence, a certain likelihood of being in a predicted location is provided. Towards building efficient prediction methods for these users, their mobility data must be studied with more sophisticated approaches.

Some studies have focused on using a set of features to obtain a good prediction performance. For instance, in [25], [26], a variety of hand-crafted features have been used. Such models lack the capability of capturing semantic information from people's mobility data. To overcome this drawback, a deep learning model has been utilized for automatically learning the best internal representations of the space and the time features.

In this paper, we propose to leverage RNN for modeling people's movement behaviour in order to predict their next location. Space and time are included to the network as features where their internal representation are learned by the network itself rather than relying on man-made representation. The space represents a specific location that has been visited by the user while the time represents the location visiting time. Before building the prediction model, the previously collected GPS points are converted into a sequence of interest points that represent series of locations visited by the user. In the training process, the trainable input features will be feed-forwarded into the hidden layer, together with previously accumulated hidden state. Our experiments on a large real life mobility dataset from Geolife project reveal that, such RNN structure will enhance the model effectiveness in comparison with the stateof-the-art models such as NN and Markov-based methods.

The main contributions of this paper are as follow. First, we use a look up table layer to discover adequate internal representations of the space and time input features while avoiding man-made representations. Second, we use RNN to model people movement and successfully incorporate the recurrent structure with space and time features into enhancing the model efficiency. Third, an extensive set of experiments is conducted on a large real life mobility dataset in order to evaluate the efficiency of the developed model. The experiments show that the RNN model outperforms the state-of-the-art models by up to 26% in terms of Recall@N performance metric. STF-RNN is implemented using Theano [23]. We plan to make the source code of STF-RNN model publicly available to be used by other researchers.

The rest of this paper is organized as follows. Section 2 provides a brief state-of-the-art on the models employed in mobility studies to predict people next locations. Section 3 explains the new proposed model defined in this study. Section 4 presents and discusses in detail the experimental results. Finally, we conclude the paper in section 5.

2. State of the art

This section summarizes different types of models that have been proposed in mobility prediction including MC, NN and RNN models highlighting the employed dataset and the techniques used for the analysis and prediction.

RNN is employed in [21] to build a mobility prediction method in wireless Ad-Hoc networks. To evaluate the efficiency of the mobility predictor, Random Waypoint Mobility (RWM) model is used to generate location time series dataset. The proposed recurrent neural network architecture has three layers, namely, input, hidden and output layers. The three layers are arranged in a feedforward connection and trained with Backpropagation algorithm. The input layer receives the data from the time series of the previous location observations as well as from the output layer feedbacks. The characteristics of the learned patterns are stored in the hidden layer. The location prediction can be used to estimate the expiration time of the links connecting the nodes enabling them to select the most stable paths which improves routing performances. Unfortunately, the author doesn't present details on the results.

Another recent work in which RNN was used is introduced in [22]. This work proposed a global prediction model called Spatial Temporal Recurrent Neural Network (ST-RNN) for predicting where users will go next. Two typical datasets called Global Terrorism Database (GTD) and Gowalla dataset are used to evaluate the effectiveness of ST-RNN model. The recurrent structure is utilized to capture not only the local temporal contexts but also the periodical ones. The spatial and temporal values are divided into discrete bins in order to produce the time-specific and distance-specific transition matrices. The corresponding transition matrix is calculated for each specific temporal value in one time bin and similarly for each specific spatial value. Unlike ST-RNN, in our model, the space and time features were fed directly into the network and the network itself is responsible for learning their internal representation.

Building local and global predictors to predict a person's next movement based on NN are proposed in [16]. Movement histories of four persons of the research group at the University of Augsburg are used to evaluate the neural predictor. In their model, they use the simplest multi-layer perceptron with one hidden layer trained with Backpropagation algorithm. The bit encoding is used to represent the rooms and the persons. In the case of local predictor, each NN is trained with the movement of a single person. Therefore, only the codes of the last visited rooms will be the input to the network. In the case of global predictor, one NN is trained with the movement of all persons. Hence, both codes of person and the last visited rooms will be fed to the network. After several experiments, the optimal configuration of the NN is determined. Two and three neurons are used in the input and hidden layer respectively. Their evaluations showed that the local predictor overcomes the global predictor with accuracy of 92.32% and 87.3% respectively.

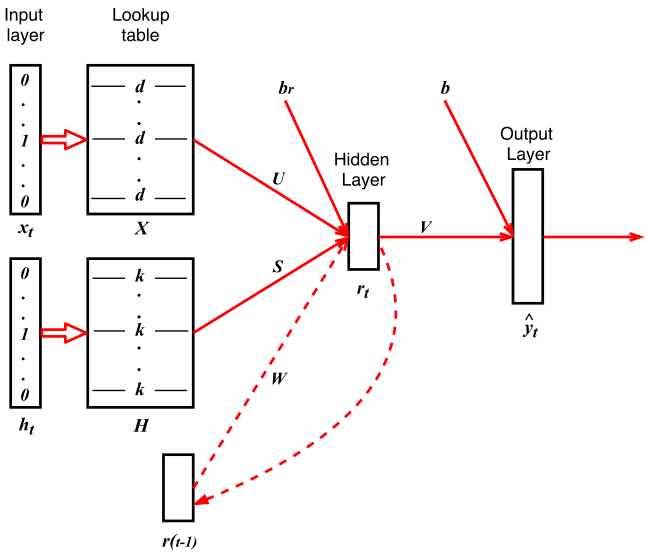


Figure 1: STF-RNN architecture.

The works introduced in [12]–[14] used NN technique to build a prediction-based location management scheme for mobile users in cellular communication network. The evaluation of the prediction scheme is conducted on dataset obtained from the MH's history of movement patterns that contains two mobiles MH1 and MH2. The movement data is represented as a pair (ds_i, dr_i) where ds_i is the distance the mobile user has crossed in cells and dr_i is the direction of movement at time i. A three-layer artificial NN trained with Backpropagation technique is used for building the prediction model. The hidden layer contains eight neurons while both input and output layers contain two neurons representing the distance and the direction. Sigmoid function is used as non-linear activation function. This prediction is used to automatically update the location information of the mobile user which reduces the traffic load in cellular communication networks. Unlike their model, where the time factor is not considered, STF-RNN has incorporated space and time interval sequences with the recurrent structure resulting in much more prediction accuracy.

The concept of n-Mobility Markov Chain (MMC) is proposed in [5]. In this approach, a track of the n visited previous locations is kept to predict a user's next location. Three different datasets are used: Phonetic, Geolife and synthetic. The datasets are collected using GPS devices for more than 180 users. Density-Joinable clustering algorithm is used to discover the Point Of Interest (POI). *n*-MMC is constructed based on the transition probabilities among the discovered POIs. Based on high frequency movement between locations, the next location is retrieved. The best model performance has been achieved when *n*=2 with 69% on Geolife dataset.

3. Methodology

This section presents data pre-processing steps. Then, the description of STF-RNN, highlighting the problem being solved in this paper, is introduced.

3.1. Pre-processing

Data pre-processing is a technique that transforms raw data into an appropriate format for further processing. We convert the GPS logs of each user into trajectories by detecting the interest points. The interest points represent those spatial regions where the user has stayed for more than a predetermined threshold providing that the distance between the start and end points of the region is under a specific threshold. For more details, please refer to [27]. These interest points will be clustered into several geospatial regions using Density-based spatial clustering of applications with noise (DBSCAN) [28]. Finally, the location history of each user is formulated using these clustered interest points.

3.2. STF-RNN: Model Description

In STF-RNN, the trajectory is represented as a sequence of tuple (x_t, h_t) where x is the centroid ID of the interest point visited at time t and h is the time unit part in hours of the leaving time from the interest point, $t = 1, 2, ..., n$, and n is the length of the trajectory. The task is to predict the future location of the mobile user at a specific time t on the basis of his/her historical mobility records.

The architecture of STF-RNN model is shown in Figure 1. It consists of four layers: input layer, lookup table layer, hidden layer (with recurrent connection) and output layer. The input layer consists of two vectors. The first one is $x_t \in \mathbb{R}^N$ which represents the centroid ID of the interest point at time stamp t. This vector is encoded using 1-of-N (or one-hot encoding) where N is the number of interest points. The second vector represents the time unit part in hours of leaving time from the interest point at time stamp t. We denote this vector by $h_t \in \mathbb{R}^M$ and it is encoded also using 1-of-M encoding technique where M is the number of different time intervals. The time intervals represent the number of hours per day in which there are 24 time intervals (hour). In the one-hot vector representation, the interest points (or leaving times) are equidistant from each other without preserving any relationship among them. The lookup table layer maps the vectors of the centroid IDs and leaving times into real value vectors. The aim of the lookup table layer is to learn a meaningful representation of the interest points and the leaving times input features. This representation enables the model to capture the embedded semantic information about user behaviour and as a consequence improving the prediction performance. Therefore, the trainable features will be used as input to further layers in the network rather than using one-hot vectors. More formally, let $X \in \mathbb{R}^{N \times d}$ be the embedded matrix that represents a set of interest points, where d is the dimensionality of the embedded vector of the interest point.

The embedded vector $xe_t \in \mathbb{R}^d$ is given by multiplying the embedded matrix X and the input vector x_t.

$$xe_t = x_t X \tag{1}$$

Similarly, the embedded vector $he_t \in \mathbb{R}^k$ is given by multiplying the embedded matrix $H \in \mathbb{R}^{M \times k}$ and the input vector h_t. Here, H represents a set of leaving time and k is the dimensionality of the embedded vector of the leaving time.

$$he_t = h_t H \tag{2}$$

The objective of the hidden layer $r_t \in \mathbb{R}^{dr}$ is to maintain the user movement history where dr is the dimensionality of the hidden layer vector. The output layer $\hat{y}_t \in \mathbb{R}^N$ produces a probability distribution over the interest points and it has the same dimensionality of the input vector x_t. The values of the hidden layer and the output layer are computed as below:

$$r_t = f\left(xe_t U + he_t S + r_{t-1} W + b_r\right) \tag{3}$$

$$\hat{y} = g(r_t V + b) \tag{4}$$

In equation 3, $U \in \mathbb{R}^{d \times dr}$, $S \in \mathbb{R}^{k \times dr}$ are the weight matrices between the input and hidden layers, $W \in \mathbb{R}^{dr \times dr}$ is the recurrent connection propagating sequential signals and $b_r \in \mathbb{R}^{dr}$ is the hidden layer bias. In equation 4, $V \in \mathbb{R}^{dr \times N}$ represents the weight matrix between the hidden and output layers and $b \in \mathbb{R}^N$ is the output layer bias. The $tanh$ is used as the non-linear activation function for the hidden layer and the softmax function is used for the output layer.

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{5}$$

$$g(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{6}$$

The current input layer, as well as the previous state of the hidden layer, is used to compute the next state of the hidden layer. Thus, the next location prediction depends on not only the current input location, but also the sequential historical information. This property of RNN structure helps the model to keep track of user movement history and discover meaningful dependencies and as consequence, enhancing the model performance.

3.3. Learning Algorithm

In this section, the learning process of STF-RNN model with the Backpropagation through time (BPTT) algorithm is presented. As stated before, the next state of the hidden layer is computed based on the current input layer as well as the previous state of the hidden layer. The cost function used in this work is the cross entropy which is defined as:

$$J = -\sum_{i=1}^{n} y_i log(\hat{y}_i) \tag{7}$$

where n is the number of training samples, y is the real user' next location, and $\hat{y}$ is the predicted next location probability. Because we have represented the centriod IDs of the interest points using one-hot vector representation, the cost function can be re-defined as:

$$J = -log(\hat{y}_i) \tag{8}$$

AdaDelta [29] is employed to estimate the model parameters, $\theta = [X, H, U, S, W, V, b_r, b, r_0]$, where r_0 is the initial vector for the recurrent layer. This process is repeated iteratively until reaching the convergence state.

4. Experiments and Results

In this section, large-scale experiments are conducted to validate STF-RNN model's effectiveness. The settings of the conducted experiments and also the dataset used for the evaluation are described first. Then, a description of the evaluation procedure and the performance metric are introduced. Finally, the experimental results are reported in detail.

4.1. Dataset

We evaluate STF-RNN model using a large real life GPS trajectory dataset belongs to Geolife project (Microsoft Research Asia) [30]. The dataset was collected by 182 users in a period of over five years (from April 2007 to August 2012). This dataset contains 18,670 trajectories with a total duration of 50,176 hours and a total distance of 1,292,951 kilometers. The collected dataset covers different cities located in China, USA and Europe but the majority of the data was from Chinese cities. These trajectories were recorded by different GPS-phones and GPS loggers every $5 \sim 10$ meters or every $1 \sim 5$ seconds. The GPS trajectories are represented by a sequence of time-stamped points, each of which contains the dimensions of latitude, longitude, altitude and other information.

4.2. Evaluation Procedure and Performance Metric

The prediction evaluation procedure has been carried out as follows. First, the previously visited locations (represented by the centroid IDs) in each trajectory are read sequentially. Then, the next locations to be visited are predicated based on these readings. Finally, the output of the model (the predicted locations) is mapped into the real data.

We use the Recall score as an evaluation metric in all experiments in order to assess the efficiency of the prediction model. The **Recall@N** is defined as the ratio between the number of correct predictions over the total number of predictions. To compute the Recall score, first, a ranked list is populated with all potential next locations arranged in a descending order according to their probabilities. Then the Recall score is calculated as the percentage of the times in which the real next location was found in the top N most

probable locations within the ranked list. In our study, we only report Recall@N with N = 1, 2 and 3. Supposing that L_u denotes the set of correspondingly real visited locations by a user u in the test data and $P_{N,u}$ denotes the set of top N predicted locations, the definitions of Recall@N is formulated as below:

$$Recall@N = \frac{1}{|U|} \sum_{u \in U} \frac{|L_u \cap P_{N,u}|}{|L_u|} \quad (9)$$

Where U is the set of users.

4.3. Experimental Settings

As stated before, building an accurate prediction model for all users is not easy task because the next location prediction is a user specific problem. Thus, building one prediction model for each user might be preferable. The model of each user is trained on its own mobility data using three-fold cross validation technique. The mobility data of each user is partitioned into three sub-data of equal size. The Recall score of each case from each user is then calculated and the final results of all users are averaged.

In order to investigate the model effectiveness, we have compared STF-RNN with the state-of-the-art location prediction models including:

- **MC**: It is proposed in [5] which is the most popular location prediction model.
- **NN**: This model is introduced in [15], [16] for predicting a person's next movement.
- **RNN**: It is introduced in [21] and it is widely used for time series prediction.

The goal of these comparisons is to show how incorporating the recurrent structure with the space and time interval sequences in our model has improved prediction overall performance. For precise model comparison, the common parameters of the models are given the same values. For example, number of training epochs and hidden layers in NN, RNN and STF-RNN are set to 100 and 20 respectively. For MC, the second order is use which achieved better prediction accuracy as shown in [5], [7]. The grid search method is used to evaluate various model parameters setting in order to select the optimum set of these parameters. The evaluated parameters include: the dimensionality of time-and-location embedded vector, the hidden layer size and the width of locations window (number of visited locations taken as input to the model). Finally, we got the best settings of the parameters as follows: the dimensionality of the embedded vector of the location (d) and time (k) is 100, 6 respectively, the hidden layer size (dr) is 20, and the width of location window is 2.

4.4. Results and Analysis

Table 1 illustrates the comparison between STF-RNN and the three other models mentioned previously in terms of Recall@N with different values of N 1, 2 and 3. It is clear from the table that MC and NN models have achieved approximately similar results under Recall@1. However, NN outperforms MC under Recall@2 and Recall@3. Also, it can be observed form the table that RNN outperforms both models MC and NN by up to 13.75%, and 15.31%, respectively. This can be attributed to incorporating the recurrent structure of RNN which enables the model to get more accurate results by taking into account long-term dependencies. Finally, STF-RNN has achieved the best results among all models and under all settings of Recall parameter. More specifically, in term of Recall@1, STF-RNN outperforms MC, NN and RNN by up to 24.45%, 26.16% and 9.4% respectively. The superiority of STF-RNN over MC and NN also can be attributed to incorporating the recurrent structure in the model whereas its superiority over RNN is due to the fact that STF-RNN uses the time feature in the model operations which impact positively the efficiency of the model. Also the internal representations learning of the input features enable the model to extract the embedded semantic information about the users's behaviour more efficiently.

	Recall@1	Recall@2	Recall@3
MC	0.589 (24.45%)	0.787 (11.44%)	0.87 (7.01%)
NN	0.581 (26.16%)	0.821 (6.82%)	0.911 (2.2%)
RNN	0.67 (9.4%)	0.865 (1.39%)	0.927 (0.43%)
STF-RNN	**0.733**	**0.877**	**0.931**

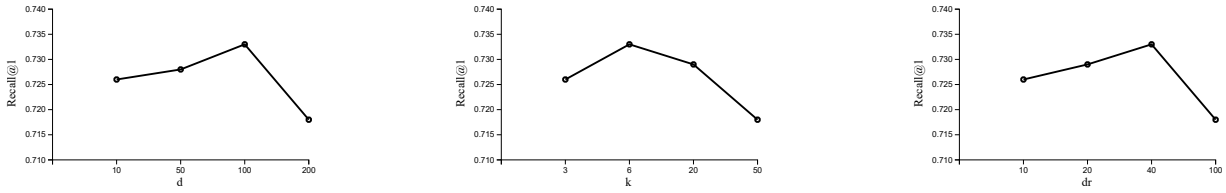
TABLE 1: Performance comparison on the dataset evaluated by Recall@N. Best scores are in bold. The values in the brackets refer to the improvement percentage of STF-RNN compared to the respective models.

4.5. Impact of Parameters

Table 2 demonstrates how varying window size may affect the performance of STF-RNN in order to determine the best setting for this parameter. The best performance is obtained with a window size value of 2 under all metrics. This is similar to the case of second order in MC which achieved the best performance as shown in [5], [7].

In STF-RNN, the parameters d, k and dr are responsible for determining the dimensionality of the embedded vectors of the location, time and hidden layer respectively so they play an important role in the efficiency of the model. To investigate the impact of these parameters and to select the best settings of them, we conduct several experiments to check the performance of STF-RNN with various dimensionalities as shown in Figure 2. We start by varying the value of one parameter while fixing the others and then studying how the model efficiency is affected. The same procedure is then repeated for the rest of the parameters.

The impact of d parameter on the Recall@1 of the model is shown in Figure 2a. The model gives the best performance when d value is around 100. The smaller values of d means that less location information is provided to the model limiting its efficiency in discovering dependencies and as consequence impairing its performance. When d value is



(a) Impact of parameter d. (b) Impact of parameter k. (c) Impact of parameter dr.

Figure 2: Parameters impact.

window	Recall@1	Recall@2	Recall@3
1	0.719	0.874	0.929
2	**0.733**	**0.877**	**0.931**
3	0.686	0.845	0.896

TABLE 2: Performance of STF-RNN evaluated by Recall@N with varying window size.

large (e.g., greater than 100), more noisy information has to be considered by the model which leads to poor performance. The effect of k and dr parameters on the model performance is depicted in Figure 2b and 2c respectively. As shown in the figures, the best performance of STF-RNN is obtained with k value of 6 while it reaches its peak under the dr parameter value of 40. Here, the fact that the model achieves its best performance with a small k value gives the impression that only a little time information is needed in representing the model dependencies in contrast with the location features where much more information is needed. The results confirm that the previous parameters play an important role in building an accurate location prediction model based on RNN.

5. Conclusions

In this paper, a Space Time Features-based Recurrent Neural Network (STF-RNN) is proposed to predict the future state of people movement. A look up table layer is used to effectively discover adequate internal representations of space and time input features enabling the model to capture the embedded semantic information about the users's behaviour more effectively. The recurrent structure is incorporated with space and time interval sequences in order to discover long-term dependencies which increases the efficiency of the proposed model. A performance evaluation is conducted on a large real life mobility dataset from Geolife project showing that our model has improved the prediction effectiveness in comparison with the state-of-the-art models.

References

[1] A. Solanas, C. Patsakis, M. Conti, I. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. Perez-Martinez, R. Di Pietro, D. Perrea, and A. Martinez-Balleste. Smart health: A context-aware health paradigm within smart cities. *IEEE Communications Magazine*, 52(8):74–81, 2014.

[2] F. Casino, J. Domingo-Ferrer, C. Patsakis, D. Puig, and A. Solanas. A k-anonymous approach to privacy preserving collaborative filtering. *J. Comput. Syst. Sci.*, 81(6):1000–1011, 2015.

[3] F. Asgari, V. Gauthier, and M. Becker. A survey on Human Mobility and its Applications. (June), 2013.

[4] Z. Vranova and V. Ondryhal. Utilization of Selected Data Mining Methods for Communication Network Analysis. *Radio engineering 20*, 2(June):548–558, 2011.

[5] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez Del Prado Cortez. Next place prediction using mobility Markov chains. *Proceedings of the First Workshop on Measurement Privacy and Mobility MPM 2012*, pages 1–6, 2012.

[6] Akinori Asahara, Kishiko Maruyama, Akiko Sato, and Kouichi Seto. Pedestrian-movement prediction based on mixed Markov-chain model. *GIS '11 Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 25–33, 2011.

[7] Wenhao Huang, Man Lit, Weisong Hut, Guojie Song, and Kunqing Xie. Hierarchical Destination Prediction Based on GPS History. pages 972–977, 2013.

[8] Wesley Mathew, Ruben Raposo, and Bruno Martins. Predicting future locations with hidden markov models. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 911–918, 2012.

[9] Conor Ryan and Kenneth N Brown. Occupant Location Prediction Using Association Rule Mining. pages 23–28, 2012.

[10] L. Chamek M. Lalam S. Hamrioui M. Daoui, M. Belkadi and A. Berqia. Mobility Prediction and Location Management Based on Data mining. (December):1–4, 2013.

[11] S. P. Kedia. An object tracking scheme for wireless sensor networks using data mining mechanism. *2012 IEEE Network Operations and Management Symposium*, pages 526–529, 2012.

[12] S. Parijaa, S. K. Nanda, P. K. Sahu, and S. S. Singh. Location Prediction of Mobility Management Using Soft Computing Techniques in Cellular Network. *International Journal Computer Network and Information Security (IJCNIS)*, (May):27–33, 2013.

[13] S. Parijaa, R. K. Ranjan, and P. K. Sahu. Location prediction of mobility management using neural network techniques in cellular network. *2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT)*, pages 1–4, 2013.

[14] S. Parijaa, P. K. Sahu, S. K. Nanda, and S. S. Singh. A functional Link Artificial Neural Network for Location Management in Cellular Network. *International Conference on Information Communication and Embedded Systems (ICICES)*, page 11601164, 2013.

[15] C. Leca, I. Nicolaescu, and C. Rîncu. Significant Location Detection & Prediction in Cellular Networks Using Artificial Neural Networks. *Computer Science and Information Technology*, 3(3):81–89, 2015.

[16] J. Petzold L. Vintan, A. Gellert and T. Ungerer. Person movement prediction using neural networks. *In First Workshop on Modeling and Retrieval of Context, Ulm, Germany*, 114(4):1–12, 2004.

[17] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T. Liu. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. pages 1369–1375, 2014.

[18] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky. RNNLM - Recurrent Neural Network Language Modeling Toolkit. pages 196–201, 2011.

[19] S. Yumlu, F. Gurgen, and N. Okay. A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction. *Proc Pattern Recognition Letters*, 26:2093–2103, 2005.

[20] T. G. Barbounis and J. B. Theocharis. Locally recurrent neural networks for long-term wind speed and power prediction. *Neurocomputing*, 69:466–496, 2005.

[21] F. Kamoun H. Kaaniche. Mobility Prediction in Wireless Ad Hoc Networks using Neural Networks. *Journal of Telecommunications*, 2(1):95–101, 2010.

[22] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the Next Location : A Recurrent Model with Spatial and Temporal Contexts. pages 194–200, 2016.

[23] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., et al. Theano: A python framework for fast computation of mathematical expressions. preprint arXiv:1605.02688, 2016.

[24] G. Pollini and I. Chih-Lin. A profile-based location strategy and its performance. *IEEE Journal on Selected Areas in Communications*, 15(8):1415–1424, 1997.

[25] Long Vu, Phuong Nguyen, Klara Nahrstedt, and Björn Richerzhagen. Characterizing and modeling people movement from mobile phone sensing traces. *Pervasive and Mobile Computing*, 17:220–235, 2014.

[26] Trinh Minh Tri Do and Daniel Gatica-Perez. Where and what: Using smartphones to predict next locations and applications in daily life. *Pervasive and Mobile Computing*, 12:79–91, 2014.

[27] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-Finder: A recommender system for finding passengers and vacant taxis. *IEEE Transaction on Knowledge and Data Engineering*, 25(10): 2390–2403, 2013.

[28] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proceedings of 2nd International Conference on KDD*, 96(34): 226–231, 1996.

[29] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[30] Y. Zheng, X Xie, and W Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, 2(33):32-40, 2010.