https://link.springer.com/chapter/10.1007/978-3-030-62365-4_17

Fernandes C., Ferreira F., Erlhagen W., Monteiro S., Bicho E. (2020) A Deep Learning Approach for Intelligent Cockpits: Learning Drivers Routines. In: Analide C., Novais P., Camacho D., Yin H. (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2020. IDEAL 2020. Lecture Notes in Computer Science, vol 12490. Springer, Cham. https://doi.org/10.1007/978-3-030-62365-4_17

A Deep Learning approach for intelligent cockpits: learning drivers routines [⋆]

Carlos Fernandes[1], Flora Ferreira[2], Wolfram Erlhagen[2], Sérgio Monteiro[1], and Estela Bicho[1]

[1] Center Algoritmi, University of Minho, Portugal
estela.bicho@dei.uminho.pt algoritmi.uminho.pt
[2] Center of Mathematics, University of Minho, Portugal
wolfram.erlhagen@math.uminho.pt cmat.uminho.pt

Abstract. Nowadays an increasing number of vehicles are being equipped with powerful cockpit systems capable of collecting drivers' footprints over time. The collection of this valuable data opens effective opportunities for routine prediction. With the growing ability of vehicles to collect spatial and temporal information solving the routine prediction problem becomes crucial and feasible. It is then extremely important to advance and take advantage of the capabilities of these cockpit systems. A vehicle that is capable of predicting the next destination of the driver and when the driver intends to leave to that destination can prepare the journey in advance. Previous studies tackling the next location prediction problem have made use of Traditional Markov models, Neural Networks, Dynamic models, among others. In this work, a framework based on the hierarchical density-based clustering algorithm followed by a Long Short-Term Memory (LSTM) recurrent neural network is proposed for spatial-temporal prediction of drivers' routines. Based on real-life driving scenarios of three different users, the proposed approach achieved a test set accuracy of 96.20%, 90.23%, and 86.40% when predicting the next destination and a R^2 Score of 93.69, 79.21, and 28.81 when predicting the departure time, respectively. The results indicate that the proposed architecture can be implemented on the vehicle cockpit for the assistance of the management of future trips.

Keywords: Human mobility patterns · Next destination prediction · Departure time prediction · Deep Learning · Intelligent vehicles

1 Introduction

The popularity of human routine prediction is rising due to its significant application value and the enhanced ability of systems to collect and process informa-

[⋆] This work received financial support from European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) and national funds, through the ADI Project Bosch & UMinho "**Easy Ride: Experience is everything**" , ref POCI-01-0247 FEDER-039334, and by FCT Fundao para a Cincia e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and UIDB/00013/2020.

tion in various spatial and temporal contexts. Understanding human mobility patterns and routines help to capture human necessities, this information can be used by intelligent systems to support individual and social events. Spatial and temporal context (where and when) are key factors for describing events. Collecting these factors is essential for analyzing and predicting human routines in practical applications. Many studies have reported that individuals daily routines exhibit a high level of spatial-temporal regularity, a high probability of returning to a few highly frequented locations and a tendency to visit specific locations at specific hours [4], [24], [15], [20], [8]. For example, in the scope of this work, drivers may have very similar weekdays routines consisting of driving their children to school, driving to work, driving back to pick their children at school, driving to the gymnasium to exercise and drive back home at the end of the evening. A daily routine is also typically dependent on temporal constraints such as going to the church every Sunday. Recently, human routines and mobility patterns have been investigated by experts in various fields such as economics [11], automotive [6], urban computing [7], criminology [16], among others. There are several possible uses for systems able to predict human routines. In terms of automotive systems, the predictions might be used to prepare the car before the driver leaves to the next destination, the predictions could also be used alongside traffic estimation systems to suggest the best driving routes to the next destination, all of this without requiring input from the driver.

In this work, we propose an efficient driving routine prediction framework based on a hierarchical density-based clustering algorithm followed by a deep neural network (DNN). Our approach focuses on building a deep learning framework through the observation of past driving scenarios using real-life data based on GPS coordinates and temporal information from three different drivers. The developed framework can be used to predict the driver's next destination and departure time to that destination.

2 Background

2.1 Clustering approaches for points of interest extraction

Some points of interest (POI) might be quite predictable and easy to extract due to the high level of repetition occurring in a driver's daily routine, for instance, places like home or work. Other POIs might be more difficult to extract due to the low level of repetition, these might be places like the museum or the cinema. Clustering techniques including K-Means clustering, hierarchical clustering, and density-based clustering are commonly applied for POI extraction [26], [13], [25]. The most popular clustering technique due to its capability of detecting clusters of varying shapes and sizes is density-based spatial clustering (DBSCAN) [22], [5]. However, the performance of DBSCAN is strongly correlated with two parameters - radius and minimum neighbors - both are always set with empirical values [27], [13]. To obtain more meaningful clusters the hierarchical clustering algorithm provides a more intuitive way for clustering exploration, additionally, various studies have shown that urban areas and human movements present a hierarchical nature meaning that hierarchical clustering approaches have a strong theoretical foundation in human routine analysis [21], [19], [1]. Hierarchical density-based clustering (HDBSCAN) is a recent clustering method that seeks to integrate both hierarchical clustering and density-based spatial clustering, this algorithm has been emerging with great success [3], [17], [14]. A driver's routine presence in some locations may be dense but dispersive in others so, using DBSCAN with empirical and fixed values might lead to poor POIs extraction since this method shows weakness when identifying clusters of different density. Contrarily, hierarchical density-based clustering can achieve better performance in POIs extraction when handling with varying density data. Therefore, HDBSCAN will be used to identify POIs in this work.

The performance of the HDBSCAN algorithm relies on the few implicit assumptions it makes about the clusters. This algorithm looks for regions of the data that are denser than the surrounding space and assumes that noise is present in the data. This clustering algorithm relies on one main parameter called m_{pts} which indicates the minimum number of points that a region must have to be considered a cluster [3].

2.2 Deep Neural Networks

Recently, the deep learning field has received a lot of attention and various DNNs have been applied successfully in multiple industries [9]. Among the several DNNs architectures recurrent neural networks (RNNs) are the most widely used when tackling temporal dynamics and time-series data. However, vanilla RNN architecture suffers from learning problems when tackling long sequences of data due to exploding gradients. This limitation was solved by developing several RNN architectures. The most successful architecture is known as long short-term memory (LSTM) [12]. The latter has been applied to analyze temporal dynamics and sequential structured data for various applications and industries such as speech recognition, natural language processing, finance, medical, among others [10], [23], [18]. The LSTM architecture is based on a set of recurrently connected cells, each cell contains one self-connected memory state, known as the internal cell state C^t, and three multiplicative units, the input, output and forget gates. These provide write, read and reset operations, and also allow the internal cell state to store and access information over long periods of time mitigating the vanishing gradient problem of vanilla RNN [12].

3 Methodology

3.1 Problem Definition

The fundamental assumption behind this work is that driving is a routine and drivers' routines show a high degree of temporal and spatial regularity. So, if a DNN can recall and learn from past routines the DNN might be able to predict the drivers' future intent.

It is important to note that perfect predictions are not realistic due to the nature of life, even if a driver has a very strict set of daily routines it is still possible for the driver to occasionally make deviations. For instance, a driver might always go to work during the weekdays at 8 a.m., however, a day will eventually come when the driver might not be able to go to work due to an illness, a car malfunction, among other factors. This rationale indicates that the prediction of drivers' routines must be a probabilistic approach. For that reason, the developed framework will analyze past routines using a hierarchical density-based clustering algorithm to identify points of interest (POIs) in the driver routine, afterwards, the DNN will be trained based on the past data and identified POIs and lastly, the outputs of the DNN will be the probability for each of the identified POIs as being the next likely destination. The DNN will also output the departure time to the next destination. When evaluating the proposed approach due to the probabilistic nature of human routines the Top-n accuracy will be considered when evaluating the next destination classification task.

This work aims to predict the next destination that a driver will visit at a specific future time based on the driver's past routine. To introduce the proposed framework some basic definitions are clarified as follows:

- **Definition 1:** A driver's routine is composed of the trip information, a single record R can be expressed as:

$$R = (c, e, t, d) \tag{1}$$

  where c denotes the car coordinates (longitude and latitude), e denotes if the car engine was turned On or Off, t denotes the time in minutes and d denotes the day of the week.

- **Definition 2:** An unlabelled routine sequence (URS) is a driver's time-ordered sequence of records which can be defined as:

$$URS = < R_0, R_1, ..., R_k > \tag{2}$$

  where k denotes the length of the unlabelled routine sequence.

- **Definition 3:** A POI is a destination that has been visited by a driver with significant frequency. A driver can have multiple POIs, these create a collection which is defined as:

$$Collection = \{POI_0, POI_1, ..., POI_n\} \tag{3}$$

  where n denotes the number of POIs.

- **Definition 4:** A driver's labelled routine item (LRI) is the combination of a record and its corresponding POI, which can be defined as:

$$LRI = (R, POI) \tag{4}$$

- **Definition 5:** A series of labelled routine items make up the driver's labelled routine sequence (LRS), which can be defined as:

$$LRS = < LRI_0, LRI_1, ..., LRI_m > \tag{5}$$

where m denotes the length of the labelled routine sequence.

3.2 Framework of the proposed approach

Fig. 1 illustrates the working steps of the proposed driver routine prediction framework. In this work, the car trip information consists of the GPS coordinates, the current day of the week, the current time in minutes (where the starting time t=0 minutes is at 0:00 a.m. of each day) and information about the current car engine state (On/Off). The framework outputs the most likely next destination and the departure time of the driver to the next destination. Learning occurs implicitly, the driver does not need to be asked for his/her regular destinations or departure schedule. The framework works as follows, firstly the data is cleaned and divided into training data, validation data, and test data. Afterward, the training data is feed into the clustering algorithm. The clustering algorithm will identify the POIs visited by the user in the past and creates the training driver routine sequence. The validation and test data are then fed into the clustering algorithm creating the respective routine sequences. The training and validation data are then used to train and fine-tune the DNN. The DNN is composed by an LSTM layer and two dense layers, the role of the LSTM layer is to extract a meaningful representation of the input routine sequence, this layer extracts higher-level features that are then fed into the dense layers to produce the two outputs: next destination and departure time to that destination. The loss functions defined for the next destination and leave time outputs are cross-entropy and mean squared error, respectively. Since the network is trained using gradient descent which minimizes a scalar, the loss functions are combined and the gradients are calculated based on the sum of the loss functions.

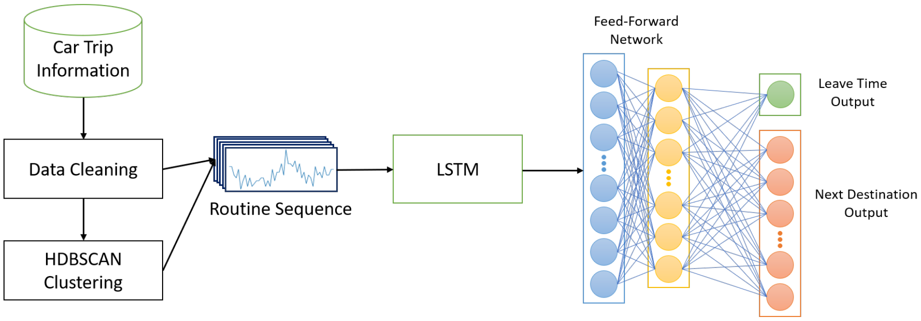


Fig. 1. Proposed Deep Learning framework.

The training of the deep learning model was performed on GPU NVidia 1060GTX with TensorFlow, to discover the optimal hyperparameters that lead to the best performance the deep learning model was trained using a randomized search method. This method consists in training the model using a defined number of combinations of hyperparameters, the hyperparameters are chosen randomly from specified distributions [2]. The hyperparameters combination that achieves the best validation performance is considered the optimal combination. To evaluate the framework Top-n accuracy and F1 score were the defined metrics for evaluating the performance of the framework when predicting the next destination (classification task). The metrics used to evaluate the performance of the framework when predicting the departure time (regression task) were mean squared error (MSE) and coefficient of determination R^2. The MSE metric is given by $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where y_i are the true values and $\hat{y}_i$ are the predicted values. For the MSE metric lower scores are preferred. The R^2 metric measures the percentage of variance in the dependent variable that the model explains and is given by $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$, where y_i are the true values, $\hat{y}_i$ are the predicted values, and $\bar{y}_i$ is the mean of the sample. The best possible R^2 score is 1.00.

4 Experiments and Results

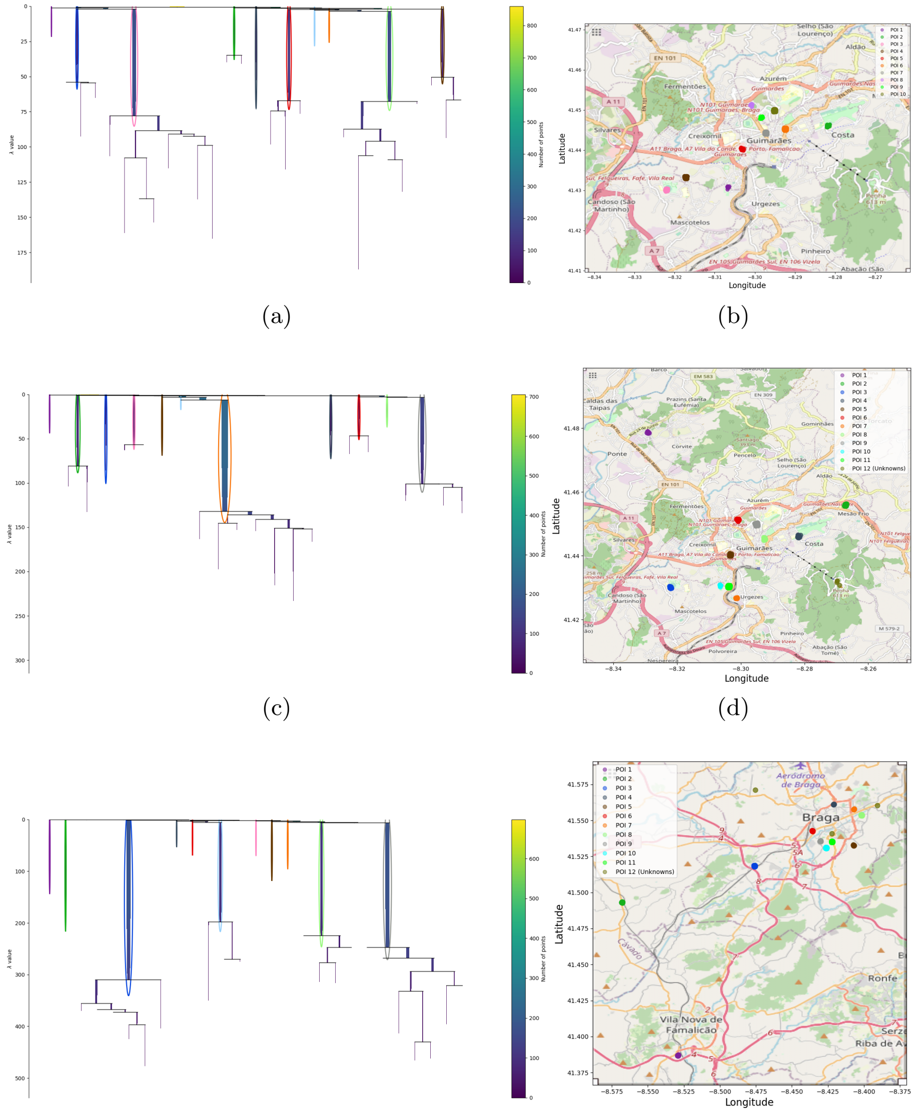
To validate the proposed framework, datasets from three different drivers were considered. A simple mobile application was developed and installed on the driver's Android phones. The volunteers were asked to press a button in the mobile application when they turn the car on or off. When the button is pressed information about GPS coordinates, date, and time were recorded. The experiment lasted for 10 weeks. Because the car is turned off and on at the same place each dataset must be a sequence of multiple two very similar records in terms of GPS coordinates (very similar because of GPS noise), this must hold true except for the first and last dataset records. Taking this reasoning into consideration, a record can be considered missing if it reflects an isolated GPS record different from the previous and the next records (this might happen due to a read-out error or a driver forgetting to use the application). In the data cleaning process, these isolated records are removed.

The data clustering was performed with the m_{pts} parameter set to 4, meaning that a destination is only considered a POI if the driver has been at that destination at least 4 times, destinations with less than 4 visits are considered as unknown. We hypothesize that clustering less-visited destinations as unknown locations will help the deep learning model to learn the underlying routine structure. The choice of setting the m_{pts} to 4 was based on the rationale that meaningful destinations of a driver are visited regularly, however this parameter can be personalized for different driver's depending on their routine.

The routine of the first driver reveals 10 POIs (Fig. 2(a) and (b)). The two most visited destinations represent 47.67% of the places the driver visits and the four most visited destinations represent 75.11%. Given this information, the first driver seems to have a strict routine without many deviations. The routine of the second driver reveals 11 POIs (Fig. 2(c)) plus 1 unknown destination (labelled POI 12, Fig. 2(d)). The two most visited destinations represent 53.82% of the places the driver visits and the four most visited destinations represent 72.52%. The second driver seems to have a strict routine but with more deviations than the first driver given the higher number of destinations. Lastly, the routine of the third driver reveals 11 POIs (Fig. 2(e)) plus 1 unknown destination (labelled POI 12, Fig. 2(f)). The two most visited destinations represent 51.58% of the places the driver visits and the four most visited destinations represent 81.37%. The third driver also seems to have a strict base routine of 4 destinations and deviations between the other less visited 8 destinations, contrarily to the second driver the unknown destinations of this driver are not close to each other.

To evaluate the performance of the framework the datasets were divided into training, validation, and test sets. The first six weeks of data were used for training purposes, the 7th and 8th weeks were used for validation and the 9th and 10th week were used for the testing phase.

Using a hyperparameter optimization technique known as randomized search, around 1500 combinations of different hyperparameters were considered to search for the best DNN performance. The considered hyperparameters were: number of hidden LSTM nodes (ranging from 32 to 100), number of hidden FFN nodes (ranging from 32 to 100), learning rate (ranging from 0.001 to 0.1), batch size (ranging from 8 to 32), FFN hidden nodes activation functions (ReLU, SeLU, tanh, sigmoid) and hidden nodes dropout rates (ranging from 0.0 to 0.5). The best combination of hyperparameters was selected based on the validation set performance, the best DNN was then used to make predictions on the test set. The performance of the best DNN for each driver is summarized in Table 1. From the analysis of Table 1, it can be concluded that the DNN is able to capture the underlying routines of each driver. The classification performance is generally very good, for the first and second drivers the training performance is slightly better than the validation and test performances, this might be a signal of over-fitting however, it might also indicate that these drivers had a slight change on their routine on the last month of the collected data. The classification performance was the lowest when tackling the third driver, this driver might have the routine with most deviations since it also was the only driver with multiple unknown destinations far from each other, nevertheless, the DNN was able to achieve a very good top-1 accuracy of 86.40% on the test set. Overall, the DNN shows very good performance when predicting the next destination of the drivers. If we consider the 3 most probable predictions of the model (Top-3 accuracy) the accuracy is 100%, 98.50%, and 94.40% on the test set of the first, second, and third driver, respectively. In terms of the departure time prediction the DNN achieved a wide range of performances, when tackling the routine of the first driver the DNN was able to understand very well the complex dynamics of departure time, it achieved a very good R^2 Score of 93.69 and 0.0059 MSE on the test set. The DNN struggled the most when tackling the routine of the third driver, the DNN was able to achieve a reasonably R^2 Score of 28.81 and 0.0486 MSE on the test set.



(a) (b)

(c) (d)

(e) (f)

Fig. 2. Clustering results obtained by HDBSCAN algorithm with the number of minimum points for cluster equal to 4. Condensed tree for Driver 1 (a), Driver 2 (c), and Driver 3 (e) and respectively representation of the points of interest for Driver 1 (b), Driver 2 (d), and Driver 3 (f).

The approach proposed in this work can achieve very good performance when predicting future destinations and departure time of drivers. However, this work has some limitations that should be addressed in future experiments. One limi-

Table 1. Performance metrics for the proposed deep learning approach on the training, validation and test sets.

	Training Set			**Validation Set**			**Test Set**		
	Driver 1	Driver 2	Driver 3	Driver 1	Driver 2	Driver 3	Driver 1	Driver 2	Driver 3
Top-1 Acc.	98.43%	95.37%	86.57%	93.84%	85.95%	85.95%	96.20%	90.23%	86.40%
Top-2 Acc.	100%	99.54%	93.98%	98.63%	92.56%	96.69%	99.46%	97.74%	92.00%
Top-3 Acc.	100%	100%	96.06%	100%	97.52%	99.17%	100%	98.50%	94.40%
Top-4 Acc.	100%	100%	99.07%	100%	100%	100%	100%	100%	97.60%
Top-5 Acc.	100%	100%	99.07%	100%	100%	100%	100%	100%	98.40%
F1 Score	95.76%	91.25%	62.65%	83.09%	73.27%	67.58%	92.41%	72.91%	71.90%
R^2 **Score**	92.41	73.55	53.74	91.27	54.31	38.37	93.69	79.21	28.81
MSE	0.0063	0.0241	0.0173	0.0075	0.0483	0.0439	0.0059	0.0195	0.0486

tation of this work is the reduced number of datasets and the duration of data collection (10 weeks). We have shown that the developed framework is able to capture weekly routines but we cannot state that monthly routines are well captured, in the future it would be beneficial to evaluate the framework with larger datasets. Another limitation of this work is that the dataset has the underlying assumption that the driver is always the same, which might not be realistic since a car can be shared, for example, between husband and wife. To mitigate this limitation, future work might focus on collecting data from the vehicle that identifies the driver and feeds that information into the model, this way the model can also understand that different drivers have different routines and habits.

5 Conclusion

In this paper, we proposed a deep learning framework based on hierarchical density-based clustering and LSTMs to predict the future destinations and departure time of drivers’. The proposed approach was able to implicitly learn the complex dynamics of daily routines and achieved very good performance both when predicting the next destination and the departure time. These findings constitute a very promising basis to further develop and experiment with driving data acquired from the vehicle’s cockpit. Provided that the stated limitations can be overcome, our work opens many perspectives for future development, first by being applied to data acquired by the vehicle’s cockpit and then to be implemented in the vehicle’s cockpit to assist drivers in their daily routines. Future work will focus on starting the implementation and testing of the proposed approach on vehicles cockpits in the scope of the joint project UMinho & Bosch “Easy Ride:Experience is everything” (ref POCI-01-0247-FEDER-03933).

References

1. Bao, J., Zheng, Y., Wilkie, D., Mokbel, M.: Recommendations in location-based social networks: a survey. GeoInformatica **19**(3), 525–565 (2015)

2. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research **13**(Feb), 281–305 (2012)
3. Campello, R.J., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pacific-Asia conference on knowledge discovery and data mining. pp. 160–172. Springer (2013)
4. Eagle, N., Pentland, A.S.: Eigenbehaviors: Identifying structure in routine. Behavioral Ecology and Sociobiology **63**(7), 1057–1066 (2009)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
6. Ferreira, F., Wojtak, W., Erlhagen, W., Vicente, P., Patel, A., Monteiro, S., Bicho, E.: A dynamic neural model for endowing intelligent cars with the ability to learn driver routines: Where to go, when to arrive and how long to stay there? In: Towards Cognitive Vehicles Workshop (TCV2019), IROS2019. pp. 15–18 (2019)
7. Gao, S.: Spatio-temporal analytics for exploring human mobility patterns and urban dynamics in the mobile age. Spatial Cognition & Computation **15**(2), 86–114 (2015)
8. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. nature **453**(7196), 779–782 (2008)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
10. Graves, A., Jaitly, N., Mohamed, A.r.: Hybrid speech recognition with deep bidirectional lstm. In: IEEE workshop on automatic speech recognition and understanding. pp. 273–278. IEEE (2013)
11. Heckman, J.J., Mosso, S.: The economics of human development and social mobility. Annu. Rev. Econ. **6**(1), 689–733 (2014)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Huang, Q.: Mining online footprints to predict users next location. International Journal of Geographical Information Science **31**(3), 523–541 (2017)
14. Järv, P., Tammet, T., Tall, M.: Hierarchical regions of interest. In: 19th IEEE International Conference on Mobile Data Management (MDM). pp. 86–95 (2018)
15. Jiang, S., Ferreira, J., González, M.C.: Clustering daily patterns of human activities in the city. Data Mining and Knowledge Discovery **25**(3), 478–510 (2012)
16. Kadar, C., Pletikosa, I.: Mining large-scale human mobility data for long-term crime prediction. EPJ Data Science **7**(1), 26 (2018)
17. Korakakis, M., Spyrou, E., Mylonas, P., Perantonis, S.J.: Exploiting social media information toward a context-aware recommendation system. Social Network Analysis and Mining **7**(1), 42 (2017)
18. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R.: Learning to diagnose with lstm recurrent neural networks. arXiv preprint arXiv:1511.03677 (2015)
19. Louail, T., Lenormand, M., Picornell, M., Cantú, O.G., Herranz, R., Frias-Martinez, E., Ramasco, J.J., Barthelemy, M.: Uncovering the spatial structure of mobility networks. Nature communications **6**(1), 1–8 (2015)
20. Rinzivillo, S., Gabrielli, L., Nanni, M., Pappalardo, L., Pedreschi, D., Giannotti, F.: The purpose of motion: Learning activities from individual mobility networks. In: International Conference on Data Science and Advanced Analytics (DSAA). pp. 312–318. IEEE (2014)
21. Roth, C., Kang, S.M., Batty, M., Barthélemy, M.: Structure of urban movements: polycentric activity and entangled hierarchical flows. PloS one **6**(1) (2011)

22. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Transactions on Database Systems (TODS) **42**(3), 1–21 (2017)
23. Siami-Namini, S., Namin, A.S.: Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386 (2018)
24. Song, C., Qu, Z., Blumm, N., Barabási, A.L.: Limits of predictability in human mobility. Science **327**(5968), 1018–1021 (2010)
25. Xu, Y., Shaw, S.L., Zhao, Z., Yin, L., Fang, Z., Li, Q.: Understanding aggregate human mobility patterns using passive mobile phone location data: a home-based approach. Transportation **42**(4), 625–646 (2015)
26. Yuan, J., Zheng, Y., Xie, X.: Discovering regions of different functions in a city using human mobility and pois. In: 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 186–194 (2012)
27. Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L.: Discovering personally meaningful places: An interactive clustering approach. ACM Transactions on Information Systems (TOIS) **25**(3), 12–es (2007)