Vehicle Trajectory Prediction based on LSTM Recurrent Neural Networks

André Ip[†], Luis Irio[*], Rodolfo Oliveira[†*]
[†]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT,
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[*]IT, Instituto de Telecomunicações, Portugal

***Abstract*—This work presents an effective tool to predict the future trajectories of vehicles when its current and previous locations are known. We propose a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) prediction scheme due to its adequacy to learn from sequential data. To fully learn the vehicles' mobility patterns, during the training process we use a dataset that contains real traces of 442 taxis running in the city of Porto, Portugal, during a full year. From experimental results, we observe that the prediction process is improved when more information about prior vehicle mobility is available. Moreover, the computation time is evaluated for a distinct number of prior locations considered in the prediction process. The results exhibit a prediction performance higher than 89%, showing the effectiveness of the proposed LSTM network.**

Keywords: Trajectory Prediction, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) Network, Transportation Data Analytics, Deep Learning.

I. Introduction

Vehicle trajectory prediction has been a topic of high interest due to the vast domain of applications, including but not limited to mobility management, data traffic optimization, data placement, to the optimization of the quality of service (QoS) of vehicular and wireless networks. A useful application of trajectory prediction is the estimation of the next location where a mobile user will require a network service. Thus, the network operators on reserve resources in advance and react efficiently to the network changes. Moreover, the fifth-generation (5G) systems adopt a technology called Mobile Edge Computing (MEC) to provide computing services for vehicles, which requires accurate and efficient trajectory prediction models, to reduce the transmitting and computing latency. In this way, vehicle trajectory prediction plays a critical role in designing future mobile communication systems.

Trajectory prediction relies on estimating the next location or consecutive locations where a vehicle will move to, using prior data describing the different vehicles' mobility statistics. The prediction models available in the literature are divided into two main categories: short-term and long-term trajectory prediction models. The short-term trajectory prediction models [1] rely on one or two previous locations and on the current location to predict the next location. In contrast, the long-term trajectory prediction models [2] predict the location of the vehicles at a more distant future time. Most of the works in the literature only consider short-term models to predict vehicular trajectories in urban areas, since long-term prediction models achieve poor prediction performance due to the randomness of the vehicles' position over time [3]. The works available in the literature addressing the trajectory prediction problem usually assume regular human trajectories or patterns [4], usually limited to residential areas, making the prediction challenge more straightforward. In contrast, the regular movements of a taxi's trajectory are less deterministic due the wide diversity of taxis' journeys. In the literature the trajectory prediction models consider that each inference unit is composed of a sequence of trajectory locations to fully learn the vehicles' mobility patterns. However, most of the existing works only studied the prediction performance for a fixed length of the trajectory segments.

Motivated by the difficulty of predicting taxi's trajectories in urban areas, and the curiosity about the impact of different amounts of prior data, we propose a LSTM based neural network to estimate taxis' mobility. Firstly, we preprocess the vehicles' trajectory raw data of the real traces performed by 442 taxis running in an urban area. Then, we train the LSTM neural network with the dataset of sequences that results from the preprocessing method, adopting five different lengths of prior information. Lastly, we perform an evaluation of the proposed prediction process based on the LSTM neural network's output.

The rest of the article is organized as follows. Section II reviews the related works. Section III presents some basic terms and definitions and analyzes the dataset used in the training process. In Section IV, we describe the LSTM neural network to predict the vehicle trajectory. Section V performs an evaluation of the proposed prediction model and Section VI concludes the paper.

II. Literature Review and Contributions

According to [5], several studies have been proposed to address the problem of trajectory prediction. For example, Dash *et al.* [6] proposed a dynamic Bayesian network, which models a sequence of variables (location, day of the week, time of the day) to predict the user's next place. However, the sparsity problem of trajectories makes it challenging to conduct inference from the historical data in a Bayesian network, resulting in a computationally intensive method. More recently, the authors of [7] addressed sparse historical

trajectory data by proposing a prediction model that employs the pattern of group travelers to improve personal location prediction precision. However, a prediction model based on group trajectory can be extremely vulnerable to external crowd behavior. Under the scope of LSTM neural networks, [8] proposed an LSTM to predict vehicles' trajectory. However, this approach is mainly designed for a highway scenario and uses high diversity of features, such as location, velocity, and vehicle's type, limiting its practicability. The work in [9] extended the work [8] by employing a probabilistic vehicle trajectory prediction method based on LSTM RNNs, which estimates the future position of the surrounding vehicle at a predetermined time. However, this approach is limited because it cannot determine the continuous trajectory. Aiming to improve the prediction accuracy, Chandra *et al.* [10] modeled the interactions between the road agents through an LSTM for predicting their trajectories in real-time. However, the model is designed for dense heterogeneous traffic, having some limitations on homogeneous or sparse traffic. Similarly, but designed for a dynamic traffic environment, [11] proposed an LSTM based model, which runs on the MEC to predict the vehicles' trajectories.

Concerning to the existing literature, this paper presents the following contributions:

- The design of a LSTM neural network that predicts the vehicles' trajectory by using the prior sequence of trajectory locations;
- The pre-processing of the vehicles' trajectory raw data to train the LSTM neural network with the sequences of trajectory locations;
- The impact of different amounts of prior data considered in the LSTM neural network input is evaluated for two distinct evaluation metrics: prediction performance and computation time;
- The presentation of experimental results showing the high prediction performance of the proposed LSTM RNN and the low computation time of the prediction process.

III. Preliminaries

A. Basic Terms and Definitions

In this work, we consider multiple vehicles traveling in a specific region, represented by a grid map. The grid map is two-dimensional, and each logical geographical sub-region of the map is denoted as a cell (c_η). An example of a grid map representation can be seen in Fig. 1. The position of each vehicle is sampled periodically and associated to a cell of the grid map.

Next, we introduce several terms and definitions adopted in the rest of the paper.

Definition 1. *A **cell** c_η with $\eta \in \{1, ..., N\}$ represents each two-dimensional division of the grid map where the vehicles are moving, where N represents the maximum number of cells.*

Definition 2. *A **trajectory** $T_\kappa = \{c^1, c^2, ..., c^\Xi\}$ denotes a list of Ξ cells sequentially visited by a vehicle. Each trajectory has a variable number of Ξ visited cells.*

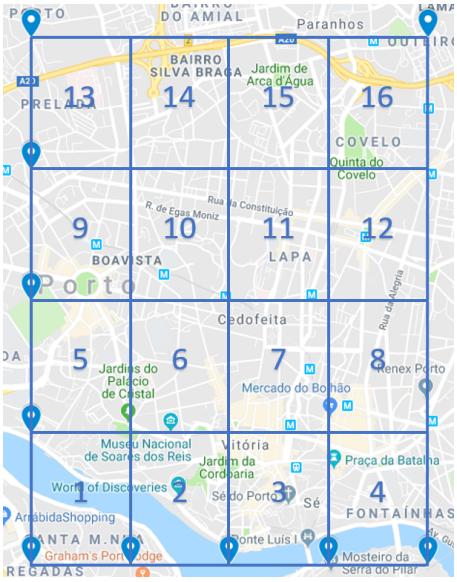


Fig. 1. Grid map representation with $N = 16$ cells.

Spatio-temporal trajectories are generated when the vehicles move from the starting point to the endpoint of a journey. Considering that each journey has a variable duration, the trajectories are divided into multiple sequences, to guarantee a coherent granular temporal basis.

Definition 3. *A **sequence** $S_\kappa = \{c^1, c^2, ..., c^\Lambda\}$ is formed by a finite set of Λ consecutive visited cells. The number of cells (Λ) that compose the sequence $\kappa \in \{1, ..., \Omega\}$ is the same for all Ω sequences. In the network training phase, a sequence is considered as a unit of inference.*

Definition 4. *The **set of sequences** $\Phi = \{S_1, S_2, ..., S_\Omega\}$ is formed by all Ω sequences that result from the preprocess of vehicles' trajectories. The set Φ will have a significant number of identical sequences. Thus, Ψ represents the number of unique sequences found in Φ.*

The definitions of the symbols are listed in Table I.

TABLE I
List of symbols.

Symbol	Description
N	Number of cells of the grid map
c_η	Cell η, $\eta \in \{1, ..., \mathrm{N}\}$
Ξ	Number of cells in a trajectory
S_κ	Sequence κ, $\kappa \in \{1, ..., \Omega\}$
Ω	Total number of sequences
Λ	Sequence length
Φ	Set of sequences
Ψ	Number of unique sequences in set Φ

B. Data Analysis

The dataset used in this work contains the real traces of trajectories performed by 442 taxis running in the city of

Porto, Portugal, during a complete year (from 01/07/2013 to 30/06/2014). Each taxi operates through a central taxi dispatch system, using GPS data acquisition devices installed in the vehicles. The dataset is provided by the UCI Machine Learning Repository, and is described in [12]. Each data entry corresponds to a completed trip of a taxi, containing a total of 9 features. Still, the ones relevant for this work are the identifier of each trip (TRIP_ID), the timestamp of the trip's start (TIMESTAMP), and the polyline containing the GPS coordinates regarding the trajectory of the trip (POLYLINE). Each polyline has a list of GPS coordinates mapped as a string in the format [longitude, latitude], acquired once every 15 seconds. The first pair represents the trip's start and the last the trip's end. Table II represents the relevant features for this work.

TABLE II
FORMAT OF EACH ENTRY DATASET.

TRIP_ID	TIMESTAMP	POLYLINE
0589	1372636858	$[[-8.618, 41.141], ..., [-8.630, 41.154]]$

We preprocess the vehicles' trajectory raw data, which includes converting each pair of GPS coordinates to a cell of the grid map and partitioning each trajectory into sequences. To simplify the process of describing trajectories, we consider that a trajectory is no longer represented by a list of GPS coordinates but as a list of cells. Thus, each pair of coordinates of a trajectory is converted into a cell of the grid map represented in Fig. 1. Having characterized each trajectory as a list of cells, we collect Λ consecutive cells to form a sequence, S_κ. The next sequence $S_{\kappa+1}$ starts with a shift of 1 cell from the beginning of S_κ. The entire process is repeated for all sequences of a trajectory and all trajectories.

In Table III, we analyze the total number of sequences (Ω) and the number of distinct sequences (Ψ), for five different Λ values ($\Lambda = \{4, 8, 12, 16, 20\}$). The total number of sequences increases as Λ decreases, meaning that more sequences are needed to characterize all trajectories when we consider fewer cells per sequence. Besides that, the number of unique sequences (Ψ) increases with Λ.

TABLE III
TOTAL NUMBER OF SEQUENCES (Ω) AND NUMBER OF DISTINCT SEQUENCES (Ψ).

Λ	Ω	Ψ
4	2752580	1349
8	2354791	10473
12	1967109	36403
16	1598190	82082
20	1263477	136450

IV. LSTM FOR TRAJECTORY PREDICTION

We address the problem of trajectory prediction of vehicles, using the $\Lambda-1$ prior observed cells to predict the next one, c^Λ. The prediction is based on an LSTM recurrent neural network due to its reputation for dealing with sequential data.

This section describes the structure of the LSTM neural network and specifies the enhancements involved in the training phase to obtain accurate vehicle trajectory predictions.

A. Design of LSTM structure

We use an LSTM recurrent neural network to solve the vanishing gradient problem that can have a significant impact on dealing with long-term sequential data [13]. The LSTM is well-known for sharing the cell states across each forward step, making the architecture ideal to deal with trajectories, where it is possible to reinforce important patterns or discard the redundant instead.

In Fig. 2, we show the structure of the adopted LSTM neural network. As can be seen, the LSTM layer is composed by $\Lambda - 1$ discrete time steps and admit an input vector $\mathbf{X}_i$ ($i \in \{1, ..., \Lambda - 1\}$) for each step, containing the information of the visited cell. We use one-hot encoding to generate the $\mathbf{X}_i$. Thus, $\mathbf{X}_i$ is a $1 \times N$ one-hot vector with the value 1 assigned to the index η, used to identify the cell of the grid map and zeros in the remaining vector. Given an input sequence represented as $\{\mathbf{X}_1, ..., \mathbf{X}_{\Lambda-1}\}$, the proposed LSTM network computes the output $\mathbf{Y}_\Lambda$. According to the input, the output is also an one-hot encoding vector ($1 \times N$ vector) where the index η that contains the single 1 will correspond to the predicted cell c^Λ, or the labeled output during the learning stage. In the LSTM layer, we adopt 16 LSTM units for each step. Since the structure of the LSTM unit may adopt different models we follow the one proposed by Hochreiter & Schmidhuber [13]. The structure of the unit is composed of the operations involving the input, output, and forget gates.

To finalize the structure, we adopt a Sigmoid function in the Dense Layer. The Sigmoid activation function is a logistic function that is useful in the prediction of one-hot vectors [13], as is the case of the desired output $\mathbf{Y}_\Lambda$.

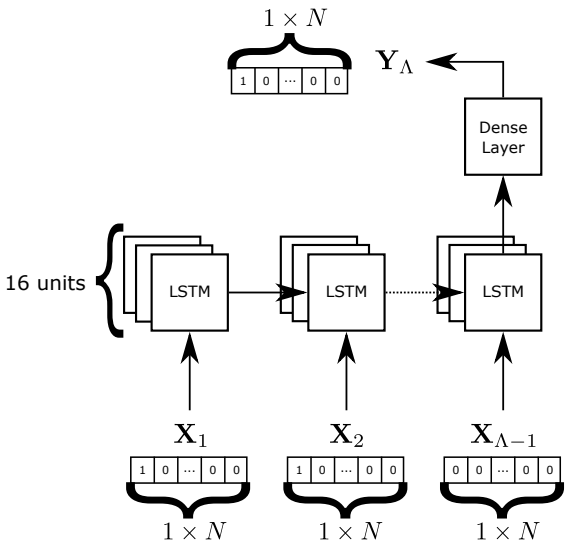


Fig. 2. LSTM structure.

B. Network Training

To learn the vehicles' mobility patterns, during the training process we use the dataset Φ containing the sequences described in the Section III-B. We selected 70% of the dataset for the training phase, and 20% for the validation. In order to optimize the training phase, we adopt the *Adam* optimizer and the categorical cross entropy as the loss function. We start the training phase with a learning rate of 0.00001, and a fixed number of 150 epochs. Furthermore, we added an early stoppage algorithm where we selected 2 levels of patience, i.e., the training phase stops when two negative oscillations in the loss function occur. We decided to add the stoppage algorithm in order to avoid over-fitting in the model.

In Table IV we present the LSTM structure and model configurations.

TABLE IV
LSTM MODEL CONFIGURATION.

LSMT Layer	Units = 16
Dense Layer	Sigmoid with 16 outputs
Loss Function	Categorical cross entropy
Optimizer	*Adam* (learning rate = 0.00001)

V. EXPERIMENT RESULTS

We performed an evaluation of the proposed LSTM network with the taxis' sequence dataset. We used the Tensorflow [14] platform with the Keras [15] library to implement the model. The LSTM network and data access procedures were built using the Python programming language. The LSTM network was trained with a NVIDIA GeForce RTX-2080 GPU, and we conducted the experiments on a Intel Core i7-8750H 2.2 GHz processor.

The validation method used to assess the prediction process is based on the output of the LSTM neural network. We evaluate the prediction performance by comparing the predicted cell c_p^Λ, with the cell c_t^Λ of the trajectory sequence in test, $S_{test} = \{c_t^1, c_t^2, ..., c_t^\Lambda\}$.

The prediction performance (PP) is the ratio of correctly predicted cells over the total number of sequences in the test set, $\mathcal{T}^{test}$, and is defined as

$$\text{PP} = \frac{1}{|\mathcal{T}^{test}|} \sum^{|\mathcal{T}^{test}|} F(c_p^\Lambda, c_t^\Lambda), \tag{1}$$

where $F(c_p^\Lambda, c_t^\Lambda)$ is 1 if the predicted cell c_p^Λ is equal the true cell c_t^Λ, and 0 otherwise.

We evaluate the prediction performance for $\mathcal{T}^{test}$, which is composed by 10^5 sequences found in Φ, and considering 5 values of Λ, thus assessing different amounts of prior data in the prediction process. From the results reported in Fig. 3, we observe that the prediction process is improved with the increase of Λ. The results show that the LSTM prediction model achieves better results as more prior observations are considered in the prediction process, i.e., by considering a LSTM RNN with more input vectors $\mathbf{X}_i$, according to the Λ value.

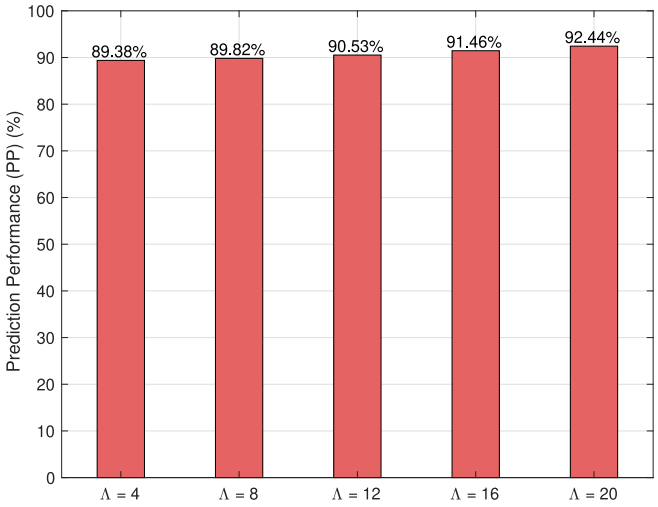


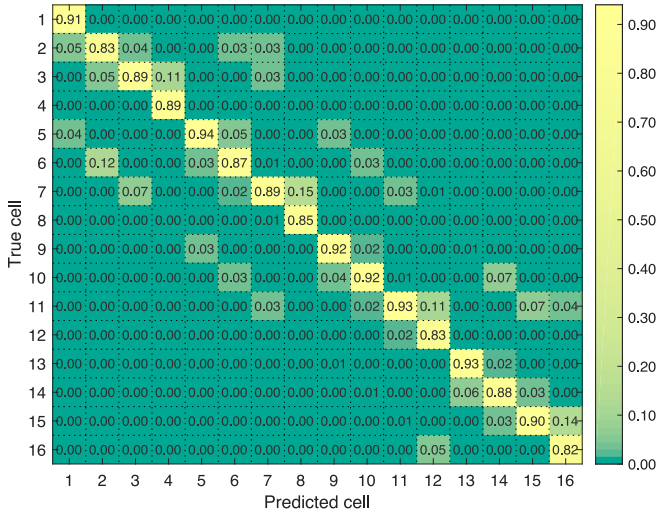
Fig. 3. Prediction performance for tested sequences.

Besides analyzing the prediction performance, we explore the confusion matrices [16] generated for $\Lambda = 4$ (Fig. 4(a)) and $\Lambda = 20$ (Fig. 4(b)). Each row represents the true (actual) cells in both confusion matrices, while each column denotes the predicted cells. The diagonal shows the cells that are correctly predicted. From Fig. 4, we observe yellow diagonals representing the high predicted performance ratios. In both confusion matrices the prediction performance of a given cell is over 82%, and the prediction method with $\Lambda = 20$ achieves a higher performance for almost all cells.

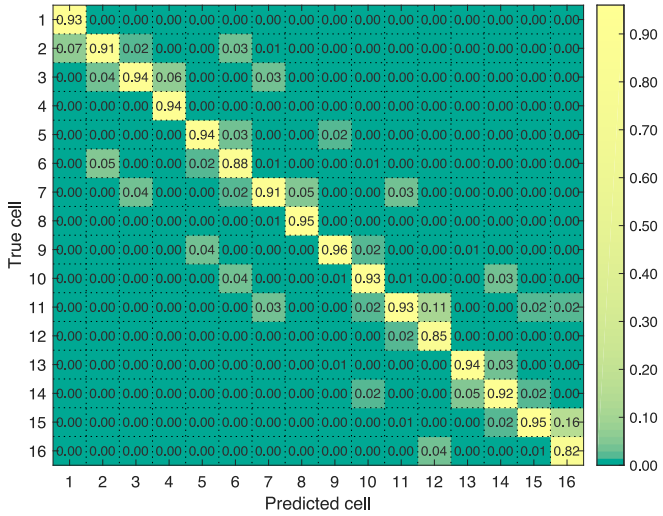
To analyze the computational efficiency of the different LSTM prediction methods, with different Λ, we evaluate the computation time for a $\mathcal{T}^{test}$ containing 1000 sequences. The computation time refers to the cumulative time to predict the next location of a number of sequences indicated on the x-axis. From Fig. 5 we conclude that the computation time increases with the parameter Λ, explained by the increasing of sequences in Φ and the number of cells per sequence. Moreover, the computation time increases linearly with the number of sequences for all LSTM prediction methods.

VI. CONCLUSION

We have proposed an accurate and efficient tool to predict the next cell where a vehicle moves to. To achieve this goal, we have adopted a LSTM RNN that is adequate to deal with sequential data. Then, we have designed an evaluation method to assess the prediction process, based on the output of the LSTM network. Experimental results demonstrate effectiveness of the proposed solution, exhibiting a prediction performance higher than 89%, and showing that the prediction process is improved when more observations are considered before performing the prediction of the next cell.



(a)



(b)

Fig. 4. Confusion matrices for two different prediction methods. (a) Confusion matrix for $\Lambda = 4$. (b) Confusion matrix for $\Lambda = 20$

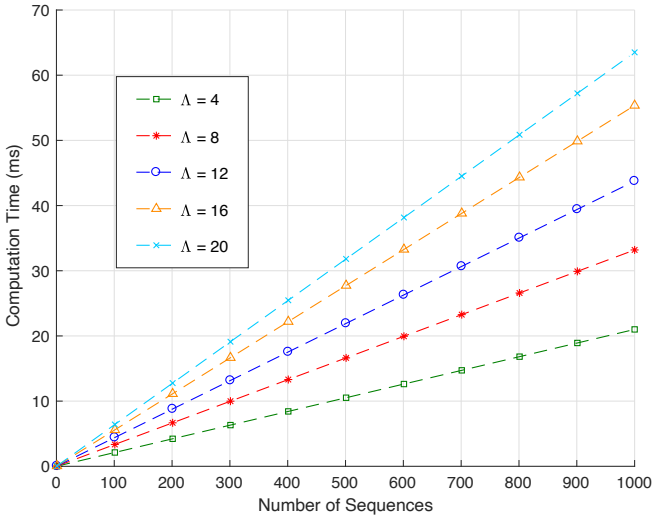


Fig. 5. Computation time of the prediction model.

Acknowledgements

This work was funded by Fundação para a Ciência e Tecnologia, under the projects InfoCent-IoT (PTDC/EEI-TEL/30433/2017), CoSHARE (PTDC/EEI-TEL/30709/2017), and Grant UIDB/50008/2020.

References

[1] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101 441–101 452, Jul. 2019.

[2] S. Qiao, N. Han, J. Wang, R. Li, L. A. Gutierrez, and X. Wu, "Predicting long-term trajectories of connected vehicles via the prefix-projection technique," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2305–2315, Jul. 2018.

[3] P. Rathore, D. Kumar, S. Rajasegarar, M. Palaniswami, and J. C. Bezdek, "A scalable framework for trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3860–3874, Oct. 2019.

[4] Y. Qiao, Y. Cheng, J. Yang, J. Liu, and N. Kato, "A mobility analytical framework for big mobile data in densely populated area," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1443–1455, Feb. 2017.

[5] H. Zhang and L. Dai, "Mobility prediction: A survey on state-of-the-art schemes and future applications," *IEEE Access*, vol. 7, pp. 802–822, Jan. 2019.

[6] M. Dash, K. K. Koo, J. B. Gomes, S. P. Krishnaswamy, D. Rugeles, and A. Shi-Nash, "Next place prediction by understanding mobility patterns," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2015, pp. 469–474.

[7] F. Li, Q. Li, Z. Li, Z. Huang, X. Chang, and J. Xia, "A personal location prediction method based on individual trajectory and group trajectory," *IEEE Access*, vol. 7, pp. 92 850–92 860, Jul. 2019.

[8] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 353–359.

[9] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 399–404.

[10] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "TraPHic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8483–8492.

[11] W. Xu, Z. Chen, C. Zhang, X. Ji, Y. Wang, H. Su, and B. Liu, "GlobalInsight: An LSTM based model for multi-vehicle trajectory prediction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[12] "Taxi Service Trajectory (TST) Prediction Challenge @ ECML/PKDD 2015." [Online]. Available: http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, 2016, pp. 265–283.

[15] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.