IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 3, 2020, accepted December 17, 2020, date of publication January 18, 2021, date of current version January 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052071

An Adaptive Learning-Based Approach for Vehicle Mobility Prediction

LUIS IRIO[1], ANDRÉ IP[2], RODOLFO OLIVEIRA[1,2], (Senior Member, IEEE), AND MIGUEL LUÍS[1,3], (Member, IEEE)

[1]Instituto de Telecomunicações, 1049-001 Lisbon, Portugal
[2]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[3]ISEL - Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, 1959-007 Lisbon, Portugal

Corresponding author: Luis Irio (l.irio@campus.fct.unl.pt)

This work was supported in part by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020, Programa Operacional Regional LISBOA (LISBOA2020); in part by the Fundação para a Ciência e Tecnologia (FCT), through projects InfoCent-IoT under Grant POCI-01-0145-FEDER-030433; and in part by the CoSHARE under Grant LISBOA-01-0145-FEDER-0307095, Grant PTDC/EEI-TEL/30709/2017, and Grant UIDB/50008/2020-UIDP/50008/2020.

ABSTRACT This work presents an innovative methodology to predict the future trajectories of vehicles when its current and previous locations are known. We propose an algorithm to adapt the vehicles trajectories' data based on consecutive GPS locations and to construct a statistical inference module that can be used online for mobility prediction. The inference module is based on a hidden Markov model (HMM), where each trajectory is modeled as a subset of consecutive locations. The prediction stage uses the statistical information inferred so far and is based on the Viterbi algorithm, which identifies the subset of consecutive locations (hidden information) with the maximum likelihood when a prior subset of locations are known (observations). By analyzing the disadvantages of using the Viterbi algorithm (TDVIT) when the number of hidden states increases, we propose an enhanced algorithm (OPTVIT), which decreases the prediction computation time. Offline analysis of vehicle mobility is conducted through the evaluation of a dataset containing real traces of 442 taxis running in the city of Porto, Portugal, during a full year. Experimental results obtained with the dataset show that the prediction process is improved when more information about prior vehicle mobility is available. Moreover, the computation time of the prediction process is significantly improved when OPTVIT is adopted and approximately 90% of prediction performance can be achieved, showing the effectiveness of the proposed method for vehicle trajectory prediction.

INDEX TERMS Trajectory prediction, hidden Markov model, estimation and modeling, machine learning.

I. INTRODUCTION

The pervasive use of mobile devices and location-based services has contributed to the production of a growing volume of spatio-temporal data, frequently exploited to improve many practical applications, ranging from the transportation engineering [1], urban planning [2], Points of Interest (POI) recommendations [3], and population distribution [4]. Furthermore, a considerable amount of mobility data has been exploited to study a variety of human dynamics [5], essential to increase the capacity of the location services used by mobile systems [6], and more recently, to forecast the spread of COVID-19 [7], [8]. In particular, GPS data-based datasets contain detailed information (e.g., latitude, longitude, and timestamp), sequentially organized to describe a trajectory [9], which is usually referred to as a set of GPS points ordered by a timestamp.

Trajectory prediction of a running vehicle based on their previous visited locations can be a key for useful applications. For instance, the prediction of the next visiting place of a user/vehicle enables advertisement companies to customize their commercial advertisements for a specific target. Moreover, trajectory prediction could help drivers to interpret the estimating travel time of a given route and perform a better route planning.

Over the past decade, different probabilistic-based models have been proposed to represent vehicles' mobility [1], [3], [10]–[12]. The models available in literature are divided in

The associate editor coordinating the review of this manuscript and approving it for publication was Camelia Delcea.

two main categories: short-term and long-term trajectory prediction models. The short-term trajectory prediction models rely on one or two previous locations plus the current location to predict the next location [13]. In its turn, the long-term trajectory models [14] predict the location at a more distant future time. However, most of existing works only employ short-term models to predict vehicular trajectories in urban areas, where the movements of vehicles are constrained by complex roads with segments, intersections and traffic. Due to the randomness of the vehicles' position over time, long-term prediction models have poor prediction performance for vehicles' mobility in cities [15]. The prediction output of long and short-term models are usually reported in the form of the next location or a sequence of future predicted locations.

Most of existing works assume that the human trajectories or patterns are very regular [16], usually limited to residential areas, making the prediction challenge more straightforward. In contrast, the regular movements of a taxi's trajectory can be reasonably rare because the customers' pick up and drop off are frequently random. In such circumstances, the prediction of the future locations can be more challenging as the level of mobility uncertainty increases.

Motivated by the difficulty of predicting highly random vehicular mobility, such as taxi's trajectories in urban areas, and the challenge of considering innovative Markov inference approaches, we propose a novel prediction methodology to estimate vehicles' mobility. In a first step, we seek to decrease the computation time and the prediction error by reducing the size of the observation state space through a pre-processing algorithm that divides a vehicle's trajectory in multiple sequences containing adequate short-time information. Then, we adopt a Hidden Markov inference model to capture the statistical properties of the taxis' mobility. The prediction approach is based on the Viterbi algorithm, which identifies the most likely sequence of locations where a vehicle will move to when a sequence of prior locations is known. Two different prediction algorithms are compared and its performance is evaluated for different model parameters.

According to the best of the authors' knowledge, no inference model has considered so far the case when the hidden states of the HMM represent a sequence of locations and not a single location. This is the main novelty of our work, as we can identify the performance of considering longer/shorter length sequences at the hidden states to evaluate the impact of the amount of prior information in the prediction process. The main contributions of this work are listed as follows:

- We propose an algorithm to convert the raw trajectory data into locations or stay-points where the vehicle stays for a period of time. The algorithm described in Section V-B provides a discretization of the mobility described by multiple GPS samples into a fixed set of locations, simplifying the computation process required for prediction;
- A novel inference model based on a Hidden Markov model (HMM) is proposed, where each hidden state represents a sequence of trajectory locations. By considering that the hidden states represent not a single location but a fixed-length sequence of consecutive locations visited by a vehicle, the HMM represents the Markov relation between sequences of prior visited locations, thus properly capturing the sequential nature of each individual trajectory;
- As an optimal solution, we present experimental prediction results computed with the Viterbi algorithm (TDVIT), which is capable of predicting the future vehicle trajectory by analyzing the previous information related to a vehicles' mobility;
- Motivated by the high of computational complexity of the optimal TDVIT algorithm as the number of hidden states increases, we propose a modified version, OPTVIT, that predicts vehicles' trajectories more efficiently without sacrificing the optimal prediction performance;
- The prediction algorithms are assessed using a dataset containing real trajectories of 442 taxis running in the city of Porto during a full year, using different evaluation metrics: computation time, and prediction performance for all unique sequences, for randomly chosen sequences, and for different sequence lengths;
- Experimental results show the impact of prior data quantity and location's resolution on both prediction performance and prediction computational time, showing that the increase of the location resolution can increase the prediction performance if longer sequences of prior data are adopted.

The rest of the paper is organized as follows. Section II reviews the related works. Section III describes the problem of vehicle mobility prediction by first setting out some basic terms and definitions. Section IV presents the methodology to infer and predict the vehicle trajectory. Section V analyzes the dataset used in this paper. Section VI describes the evaluation of the prediction process and Section VII concludes the paper.

II. RELATED WORK

This section provides a brief overview of the state-of-the-art by describing and comparing different mobility prediction schemes. Although some authors have already proposed a more detailed characterization of the existing mobility preditcion algorithms [17], the mobility prediction schemes can be divided into three categories: (i) Bayesian Network-Based Methods; (ii) Markov-Based Methods; (iii) Neural Network-Based Methods.

A. BAYESIAN NETWORK-BASED METHODS

Bayesian Network-Based methods rely on Bayesian inference to evaluate the probability of a given vehicle trajectory conditioned by an observed segment (sub-trajectory) based on statistics from historical trajectory data. Adopting a Bayesian inference, Zhang *et al.* [18] proposed a location prediction model that considers multiple predictive factors

(topology information, road topology information, and movement information) to improve the prediction accuracy and enhance the efficiency of the model. Similar to [18], the authors in [19] exploited frequent region patterns to construct a Bayesian network to predict the future location of a moving object. In a different way, Dash *et al.* [20] 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.

B. MARKOV-BASED METHODS

The most widely used mobility prediction schemes fall under the category of Markov-based methods [1], [3], [10]–[12], [21]. The Markovian characteristic refers to a sequence of possible states, where the next state depends only on the current state. Based on this characteristic, it is possible to describe the user's movement, in which the transition probability from the current to the next location is only based on the current location. Adopting a Markov-based method, the authors in [1] presented a preliminary work to predict the continuous paths of moving objects. In [21] mobility prediction is adopted for a cooperative caching scheme proposed to cache popular contents at a set of mobile nodes that stay longer at the hot spot areas. Qiao *et al.* [1] proposed an HMM in which the hidden states are trajectory segments, and the observable states are cells. However, the authors only studied the prediction accuracy for a fixed length of the trajectory segments. Further, to potentially increase the efficiency of mobility predictors, Lv *et al.* [3] studied the possibility of considering user's living habits in the HMM model, but the performance of this method presents a low accuracy of next-place prediction for users having a short history of movements or not exhibiting ergodic motion. More recently, the work in [12] addressed the problem of sparse historical trajectory data, by proposing a prediction model that employs the pattern of group travels to improve personal location prediction precision. Li *et al.* [12] firstly used the sequences of users' trajectory locations to construct a spatial clustering of multiple users' trajectories, and then adopted two variable-order Markov models to predict the clustering trajectory. However, a prediction based on group trajectory can be extremely vulnerable to external crowd behavior. Taking into account the crowd mobility, [11] proposed a HMM to perform individual trajectory prediction. Although the individual trajectory prediction can be improved through the knowledge of crowd mobility, the authors segmented the locations into limited points of interest. Thus, this method is not so adequate to predict a taxi's trajectory due to the high number of destinations compared to a few points of interest.

C. NEURAL NETWORK-BASED METHODS

Neural networks have been used for mobility prediction in different scenarios. One of the preliminary works adopting a neural network for trajectory prediction [22] won the ECML-PKDD (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases) competition [23]. As described in [22], the authors considered a Multi-Layer Perceptron with a fixed-size input. Thus, this approach can not handle input trajectories of variable length. Artificial neural networks were also adopted to predict vehicles' arrival rate that is further used to optimize the operation of vehicular routing protocols [24]. Within the scope of long short-term memory (LSTM) neural networks, [25] proposed the use of LSTMs to predict the trajectory of vehicles. However, this approach is mainly designed for a highway scenario and use high diversity of features, which limits its practicability. LSTMs were also proposed in [26] to predict personalized trajectories of connected vehicles. More recently, the authors of [27] adopted a convolutional neural network to propose a novel algorithm that predicts the destinations of taxi trajectories. Contrary to the traditional prediction models, which treats trajectories as one-dimensional sequences, the proposed algorithm models trajectories as two-dimensional images to improve the prediction accuracy. The experiments conducted on real trajectory data by Lv *et al.* [27] showed that the proposed scheme achieves better accuracy than state-of-the-art methods. However, the method needs hours to learn the spatial patterns of the trajectories.

Although Bayesian network-based methods are straightforward to implement, the sparsity of trajectories makes it challenging to conduct inference from the historical data, resulting in a computationally intensive method. Thus, the Bayesian Network-Based methods are usually combined with Neural Network-Based methods to enhance performance [13]. Regarding the Markov-based models, their performance is mainly influenced by the transition probability matrix, and the lack of extensibility associated with these models makes them very difficult to obtain such matrix when the state space is large. HMMs can improve the performance of mobility prediction due to the conditional association of hidden events, but the hidden states increase the computation complexity associated to the prediction state. Finally, the Neural Network-Based models can achieve low computation times, but for those with an extensive number of hidden layers and neurons, they demand for a long training period.

III. PROBLEM STATEMENT AND PRELIMINARIES

This section introduces some basic terms and definitions needed to define the mobility prediction model.

A. BASIC TERMS AND DEFINITIONS

In this work we consider that multiple vehicles are traveling in a specific spatial region, represented by a grid map. The grid map is two-dimensional and it is divided into geographical sub-regions denoted as cells, represented by 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.

Spatio-temporal trajectories are generated when the vehicles move from the trip's start to the trip's end. Considering that each path has a variable trip's time, the trajectories are

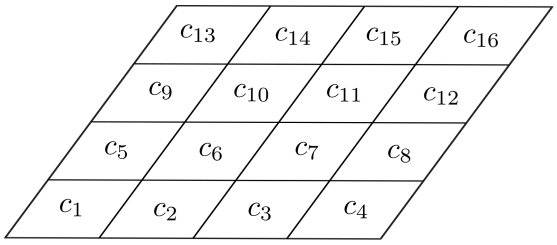

FIGURE 1. Grid map representation with N = 16 cells.

TABLE 1. List of symbols.

Symbol	Description
N	Number of cells of the grid map
c_η	Cell η, $\eta \in \{1, ..., \mathrm{N}\}$
T_j	Trajectory j
Ξ_j	Number of cells of 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
$\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$	HMM inference model
$\boldsymbol{\pi}$	Initial state distribution
$\boldsymbol{A}$	HMM Transition matrix
$\boldsymbol{B}$	HMM Emission matrix

divided into sequences that have a constant number of cells. To provide practical insight into vehicle's mobility, some basic terms and definitions are given as follows.

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

Definition 2: A ***trajectory*** $T_j = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^{\Xi_j}\}$ *represents a list of* Ξ *cells sequentially visited by a vehicle, where* c_η^k *represents the k-th cell of the trajectory. Each trajectory is composed by a variable number of* Ξ_j *visited cells, with* $\Xi_j > 1$.

To define a granular length of the trajectories' sequential data adopted in the inference model, we represent a trajectory as a set of multiple locations' subsets. Each subset is denoted as a sequence and is next defined.

Definition 3: A ***sequence*** $S_\kappa = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^\Lambda\}$ *is formed by a finite set of* Λ *consecutive visited cells, with* $\Lambda \leq \Xi_j$. *The number of cells* (Λ) *that compose the sequence* $\kappa \in \{1, \ldots, \Omega\}$ *is the same for all* Ω *sequences that jointly represent the trajectory.*

A sequence is considered as a unit of inference, since the statistics obtained during the inference stage define the probability of occurrence of each sequence and the probabilities of transition between the multiple sequences that compose a single trajectory. The definitions of the symbols are listed in Table 1.

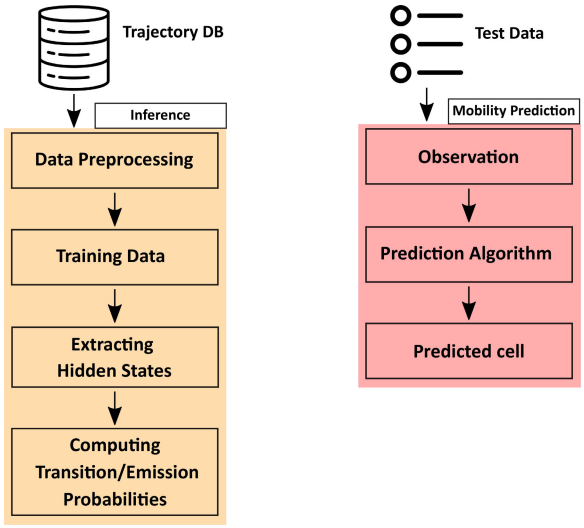

FIGURE 2. Structure of the model for vehicle mobility prediction.

B. PROBLEM STATEMENT

The structure of the proposed model is depicted in Fig. 2. At its most basic level, the model contains two main modules: 1) the statistical inference stage; and 2) the mobility prediction stage. In the statistical inference module, we obtain the historical data from the trajectory database, and then we extract and transform the trajectories to infer the stochastic properties that can be used for the mobility prediction module. Further, in the mobility prediction module, we make use of the inferred statistics to convey the motion trend and predict the mobility of the vehicles. The inference stage is based on a HMM model, presented as follows.

Definition 4 (HMM for Mobility Inference): Assume a segment of a vehicle trajectory (sub-trajectory) represented by a sequence $S_\kappa = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^\Lambda\}$, *and* $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ *a HMM, where* $\boldsymbol{\pi}$ *represents the initial state distribution,* $\boldsymbol{A}$ *is the transition matrix of hidden states, and* $\boldsymbol{B}$ *is the emission matrix of observable states.*

IV. INFERENCE AND PREDICTION METHODOLOGY

This section describes the methodology to infer the mobility statistics and to predict the vehicle trajectory based on the information inferred so far. We present a detailed overview of how the information regarding the initial state distribution, transition matrix of the hidden states, and the emission matrix of the observable states are computed. The section ends by presenting the trajectory prediction algorithms TDVIT and OPTVIT.

A. INFERENCE

We model each trajectory using a Markov chain, which consists of a set of states (represented by Ψ) and a set of transitions between them, where each transition has an associated probability. Thus, the Markov chain is defined by assigning a hidden state to each unique sequence S_κ. For each pair of

adjacent sequences $\{S_\kappa, S_{\kappa+1}\}$, the transition probability of traveling from S_κ to $S_{\kappa+1}$ is given by

$$a_{\kappa,\kappa+1} = \frac{\#(S_\kappa, S_{\kappa+1})}{\#(S_\kappa)}, \tag{1}$$

where $\#(S_\kappa, S_{\kappa+1})$ is the number of times that the two adjacent sequences $\{S_\kappa, S_{\kappa+1}\}$ occur in all trajectories and $\#(S_\kappa)$ is the number of times that S_κ appears in the inference data. Each entry $a_{\kappa,\kappa+1}$ is stored in the transition probability matrix $\boldsymbol{A}$.

In real-time, we do not observe an entire sequence but only the current position of a vehicle, represented by a cell. The HMM is a probabilistic model that comprises observed events and hidden events. In this work the observed events represent the visited cells and the hidden events are the possible sequences that characterize the user's motion. The goal is to consider a set of observed cells already visited by a vehicle (prior information) to predict the next cell to be visited.

The HMM is described by the hidden states $q(t)$ and the observable states $o(t)$ at discrete time t, and can be specified by the following components:

1) $\boldsymbol{Q} = \{q_1, q_2, \ldots, q_\Psi\}$ a finite set of Ψ hidden states, that represent the possible Ψ unique sequences;
2) $\boldsymbol{O} = \{o_1, o_2, \ldots, o_N\}$ a finite set of N observations, that represent the possible N cells of the grid map;
3) $\boldsymbol{A} = \{a_{1,1}, a_{1,2}, \ldots, a_{\Psi,\Psi}\}$ a state transition probability matrix, where each element of the matrix is given by $a_{i,j} = P(q(t+1) = S_i | q(t) = S_j)$, $i, j \in \{1, 2, \ldots, \Psi\}$.
4) $\boldsymbol{B} = \{b_1(o_1), \ldots, b_1(o_N), \ldots, b_\Psi(o_1), \ldots, b_\Psi(o_N)\}$ a emission probability matrix, where each element expresses the probability of an observation o_n being observed in the state κ, given by $b_\kappa(o_n) = P(o_n|q_\kappa)$.
5) $\boldsymbol{\Pi} = \{\pi_1, \pi_2, \ldots, \pi_\Psi\}$ is the initial state distribution, where $\pi_i = P(q(0) = S_i)$.

Fig. 3 presents an example of the HMM adopted in the inference stage, where the circles represent the hidden states, and the grid cells illustrate the observable states. In this example, the hidden state set $\boldsymbol{Q} = \{q_1 = S_1, q_2 = S_2, q_3 = S_3, q_4 = S_4\}$ corresponds to the number of unique sequences, and the observable state set $\boldsymbol{O} = \{c_1, c_2, \ldots, c_{16}\}$ contains N = 16 possible observable states, corresponding to the partitions of the grid map.

The state transition probability matrix, $\boldsymbol{A}$, and emission probability matrix, $\boldsymbol{B}$, related to Fig. 3 are given by

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0.1 & 0.1 & 0 & 0.8 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.15 & 0.85 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0.6 & 0.05 & 0 & 0 & 0.35 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \ldots & 0 \end{bmatrix}.$$

The transition probability matrix $\boldsymbol{A}$ is an $\Psi \times \Psi$ matrix and can be computed using (1). Similarly, the emission probability matrix $\boldsymbol{B}$ is an $\Psi \times$ N, where each entry $b_\kappa(o_n) = P(o_n|q_\kappa)$ is computed by

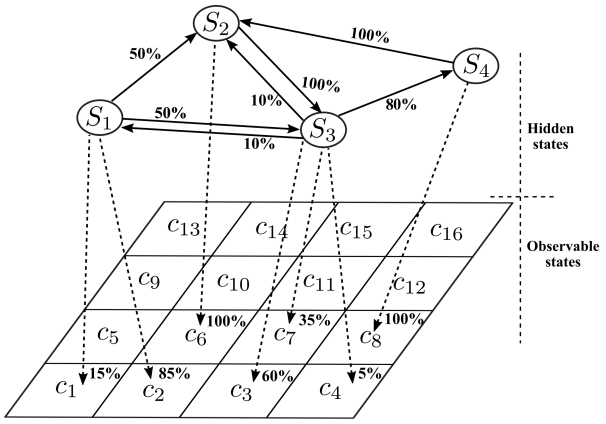

FIGURE 3. Example of an HMM for mobility inference.

$$b_\kappa(o_n) = \frac{\#(q_\kappa, o_n)}{\Lambda}, \tag{2}$$

where $\#(q_\kappa, o_n)$ represents the number of times that the observable state o_n is observed in the hidden state q_κ. $\boldsymbol{\Pi}$ is a $1 \times \Psi$ vector where each element holds the value

$$\pi_i = \frac{\#(q_i)}{\Omega}. \tag{3}$$

After the inference stage, the next goal is to estimate the most likely hidden state given a vector of observations. Next we describe the proposed prediction algorithm.

B. MOBILITY PREDICTION

The prediction problem finds the hidden state with the maximum likelihood, when previous and current information of vehicles' location is known. The prediction problem is stated as follows

$$P(\boldsymbol{O}|\lambda) = \max_{q_1 \le q_i \le q_\Psi} \{P(\boldsymbol{O}|q_i, \lambda) \times P(q_i|\lambda)\}, \tag{4}$$

where the goal is to identify the sequence of hidden states that ensures the maximum probability $P(\boldsymbol{O}|\lambda)$. The problem can be optimally solved adopting the Viterbi algorithm, which finds the chain of hidden states with the maximum likelihood. The algorithm has three phases described as follows.

1) First, the initial probability of each hidden state π_i and the emission probability of observing o_1 given the hidden state q_i are used in the initialization of the Viterbi variable $\delta_1(i)$ as follows

$$\delta_1(i) = \pi_i \, b_i(o_1), \quad 1 \le i \le \Psi. \tag{5}$$

2) Then, the forward Viterbi variable $\delta_{2:\Lambda}(j)$ is computed recursively for each hidden state q_j as the product of the previous forward variable of state q_i by the transition probability between the previous state q_i to state q_j, $a_{i,j}$, and by the emission probability from state q_j to the observable $o_{2:\Lambda}$, as follows

$$\delta_{2:\Lambda}(j) = \max_{q_1 \le q_i \le q_\Psi} \{\delta_{1:\Lambda-1}(i) \, a_{i,j} \, b_j(o_{2:\Lambda})\}. \tag{6}$$

3) Lastly, we find the Viterbi path with the maximum transition probability among all $\delta_T(i)$ Viterbi variables, as follows

$$P^* = \max_{q_1 \leq q_i \leq q_\Psi} \{\delta_T(i)\}. \tag{7}$$

In Algorithm 1, TDVIT, we adopt the Viterbi algorithm, denoted as Viterbi($\boldsymbol{O}$, λ), to predict the cell c_η^Λ of a sequence $S_{test} = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^\Lambda\}$, when the first $\Lambda - 1$ visited cells are known. We start by collecting the first $\{\Lambda - 1\}$ cells of S_{test} stored in χ (line 1). After that, we search for all unique sequences (ζ) that contain the first $\{\Lambda - 1\}$ cells equal to χ (line 2). In this manner, we have the set of all possible sequences, ζ, in which the last cell c_η^Λ is a possibility for the predicted cell. For each sequence $S_j \in \zeta$ (line 3), we gather the observable state set $\boldsymbol{O}$, that corresponds to all Λ cells of sequence S_j (line 4). Then, we use the Viterbi (line 5) to obtain the last forward Viterbi variable $\delta_\Lambda(j)$, which indicates the probability of the current sequence S_j (denoted as P^*), relatively to other candidate sequences. Next, we store the pair $\left[P^*, S_j\right]$ in vector Υ (line 6). In the last step, we find the pair $\left[P^*, S_j\right]$ with maximum probability P^* in Υ to identify the predicted sequence S_{pred} (line 8).

Algorithm 1 Prediction Algorithm TDVIT

Input: $\mathcal{T}^{test}$; $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$; S_{test}; $\boldsymbol{O}$
Output: S_{pred}
1 $\chi = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^{\Lambda-1}\} = \text{Transform}(S_{test})$
2 $\zeta = \text{Search\_sequences}(\chi, \mathcal{T}^{test})$
3 **foreach** $S_j \in \zeta$ **do**
4 $\quad \boldsymbol{O} = \{o_1, o_2, \ldots, o_\Lambda\} = \text{Transform}(S_j)$
5 $\quad P^* = \text{Viterbi}(\boldsymbol{O}, \lambda)$
6 $\quad \Upsilon.append(\left[P^*, S_j\right])$
7 $S_{pred} = \arg\max_{P^*}\{\Upsilon\}$
8 **return** S_{pred}

The time complexity of Algorithm 1 is $O(\Psi^2\,\Lambda)$. To improve the time performance of the prediction problem, we propose OPTVIT presented in Algorithm 2. The basic idea of Algorithm 2 is to compute only the first $\Lambda - 1$ Viterbi variables $\left(\delta_{1:(\Lambda-1)}(j)\right)$ for the first sequence S_j that occurs in ζ. This is made after the check in line 6, and the Viterbi variables are stored in vector V_{vit}. Considering that the first $\Lambda - 1$ observable states, $\{o_1, o_2, \ldots, o_{\Lambda-1}\}$ are the same for each $S_j \in \zeta$, we only need to compute $\delta_{1:(\Lambda-1)}(j)$ for S_1. For the remaining sequences, $S_{j>1}$, we only need to compute the last Viterbi variable $\delta_\Lambda(j)$ (line 10), based on previously Viterbi variables. Next, we store the pair $\left[P^*, S_j\right]$ in vectors Υ (line 11) and $\mathbf{R}$ (line 12). Vector $\mathbf{R}$ is essential to reduce the algorithm's time complexity since it stores the probability P^* of all computed sequences. Thus, OPTVIT only needs to calculate the Viterbi variables of sequences that have not been computed previously (after the check in line 4). Otherwise, P^* is restored from $\mathbf{R}$ (line 14). The last steps of Algorithm 2 (lines 16-17) are similar to the last steps of Algorithm 1 (lines 7-8).

We emphasize that Algorithm 1 is proposed for comparison purposes. Algorithm 2 introduces two main advantages because: (i) it is only computed the entire Viterbi path for the first sequence and then only the last cell of each possible sequence is recomputed; (ii) the Viterbi path of the first sequence is stored for future reference, so we can reuse prior computations for future sequences.

Algorithm 2 Prediction Algorithm OPTVIT

Input: $\mathcal{T}^{test}$; $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$; S_{test}; $\boldsymbol{O}$; $\mathbf{R}$
Output: S_{pred}
1 $\chi = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^{\Lambda-1}\} = \text{Transform}(S_{test})$
2 $\zeta = \text{Search\_sequences}(\chi, \mathcal{T}^{test})$
3 **foreach** $S_j \in \zeta$ **do**
4 $\quad$ **if** $S_j \notin \boldsymbol{R}$ **then**
5 $\quad\quad \boldsymbol{O} = \{o_1, o_2, \ldots, o_\Lambda\} = \text{Transform}(S_j)$
6 $\quad\quad$ **if** $j = 1$ **then**
7 $\quad\quad\quad P^* = \text{Viterbi}(\boldsymbol{O}, \lambda)$
8 $\quad\quad\quad V_{\text{vit}} = \delta_{1:(\Lambda-1)}(j)$
9 $\quad\quad$ **else**
10 $\quad\quad\quad P^* = \text{Viterbi}(o_\Lambda, \lambda, V_{\text{vit}})$
11 $\quad\quad \Upsilon.append(\left[P^*, S_j\right])$
12 $\quad\quad R.append(\left[P^*, S_j\right])$
13 $\quad$ **else**
14 $\quad\quad P^* = \mathbf{R}\left[S_j\right]$
15 $\quad\quad \Upsilon.append(\left[P^*, S_j\right])$
16 $S_{pred} = \arg\max_{P^*}\{\Upsilon\}$
17 **return** S_{pred}

V. OFFLINE MOBILITY DATA ANALYTICS

This section describes how to 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. This section also includes an analysis of the mobility dataset considered in this work, according to the proposed inference model.

A. DATA DESCRIPTION

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 [23]. 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).

TABLE 2. Format of each entry dataset.

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

TABLE 3. Area of each grid cell.

N	Area (km^2)
16	0.724
64	0.181
256	0.045

Each polyline has a list of GPS coordinates mapped as a string in the format [longitude, latitude], sampled once every 15 seconds. The first pair represents the trip's start and the last the trip's end. Table 2 represents the relevant features for this work. We emphasise that the scope of this work is merely on predicting the vehicular trajectory based on prior locations without using enriching data of other types, so we could evaluate the impact of the amount of prior information on the prediction process, without having any influence of additional enriching data.

B. DATA PREPROCESSING

The handling of trajectory raw data plays an essential role in prediction performance. In this section, we describe the data preprocessing algorithm to transform the data characterizing the trajectories. The main objective of the algorithm is to construct a set of sequences to be used in the inference stage.

The spatial region where vehicles are moving is represented by a rectangular grid, as illustrated in Fig. 1. The latitude and longitude of the lower-left corner are 41.1391696 and −8.6341313, respectively. The grid dimension is 2953 m (height) by 3921 m (width). The region is divided into N cells, as represented by the grid map in Fig. 1 (N = 16), and the area of each cell is illustrated in Table 3 for N = 16, N = 64, and N = 256.

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, c_η. Considering that each trajectory has a variable number of Ξ_j visited cells ($T_j = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^{\Xi_j}\}$), a sequence ($S_\kappa$) is used to represent a finite set of Λ cells: $S_\kappa = \{c_\eta^1, c_\eta^2, \ldots, c_\eta^\Lambda\}$. The total number of cells (Λ) that compose the sequence $\kappa \in \{1, \ldots, \Omega\}$ is the same for all Ω sequences.

Having characterized each trajectory as a list of cells, we develop an algorithm to build the sequences from raw data. The proposed algorithm is shown in Algorithm 3. The algorithm starts by verifying if the number of cells that compose the sequence S_κ is greater than 1 (line 2) guaranteeing that no sequence is formed by less than one cell. For all trajectories (line 5), the algorithm gathers a total of Ξ_j cells in each trajectory (line 7). Each collected cell is added to a given sequence S_κ, up to a maximum of Λ cells per sequence (line 9). After that, the sequence S_κ is added to the set of sequences Φ (line 11). The next sequence $S_{\kappa+1}$ (line 12) starts with a shift of 1 cell from the beginning of S_κ (line 14). The entire process is repeated for all trajectories.

Algorithm 3 Data Preprocessing

Data: Dataset of *trajectories* T_j (as a list of cells)
Input: Λ
Output: $\Phi = \{S_1, S_2, \ldots, S_\Omega\}$

1 $\Phi = \{\}$
2 **if** $\Lambda > 1$ **then**
3 $\kappa = 1$
4 $S_\kappa = \{\}$
5 **forall the** T_j **do**
6 $i = 1$
7 **foreach** $c_\eta^1, \ldots, c_\eta^{\Xi_j} \in T_j$ **do**
8 $S_\kappa.append(c_\eta^i)$
9 $i = i + 1$
10 **if** $i > \Lambda$ **then**
11 $\Phi.append(S_\kappa)$
12 $\kappa = \kappa + 1$
13 $S_\kappa = \{\}$
14 $i = i - (\Lambda - 1)$
15 **return** Φ

C. DATA EVALUATION

Here, we evaluate the set of sequences, Φ, computed in Algorithm 3. In Table 4, we analyze the total number of sequences (Ω) and the number of unique sequences (Ψ), for $\Lambda = \{4, 8, 12, 16, 20\}$ and N = 16. The total number of sequences increases as Λ decreases, which means that more sequences are needed to characterize all trajectories when fewer cells per sequence are considered. Besides that, the number of unique sequences increases for longer sequences.

Note that each unique sequence S_κ has a certain probability of occurring in the dataset. To assess the different probabilities, we plot the cumulative distribution function (CDF) of sequences' occurrence in Fig. 4. We represent each observed sequence S_κ in x-axis, starting with the most likely sequences and the cumulative probability of S_κ in y-axis. We observe in Fig. 4 that different CDFs are obtained for the five values of Λ, indicating that the slope of the CDF slowly decreases as Λ increases. Consequently, the increase of Λ is beneficial, because the vehicles' trajectories can be represented by multiple sequences with lower occurrence probability, decreasing the number of dominant sequences that might lower the prediction performance.

In Table 5, we evaluate the total number of sequences (Ω) and the number of unique sequences (Ψ), for N = {16, 64, 256} by setting $\Lambda = \{4, 20\}$ and always considering the same grid dimensions as described in Subsection V-B. We observe that Ψ increases with N for a fixed Λ, while Ω remains constant. Due to the increase of

TABLE 4. Number of sequences (Ω) and number of unique sequences (Ψ) obtained for different Λ values.

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

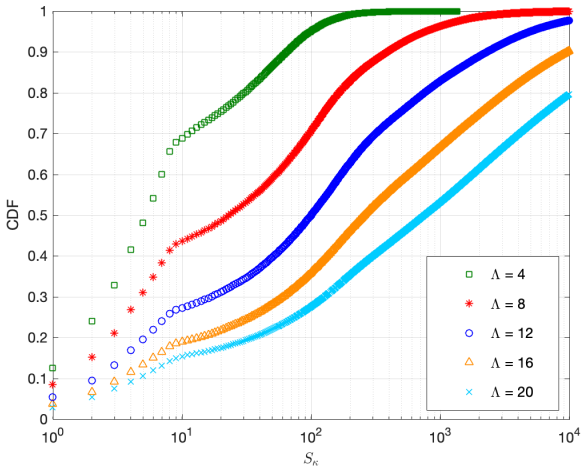

FIGURE 4. CDF of sequences' occurrence in the dataset.

TABLE 5. Number of sequences (Ω) and number of unique sequences (Ψ) obtained for different N values.

N	Λ	Ω	Ψ
16	4	2752580	1349
	20	136450	588711
64	4	2752580	5971
	20	1263477	588711
256	4	2752580	23340
	20	1263477	1048137

Ψ with N, the number of more dominant sequences also increases with N. The CDFs of sequences' occurrence represented in Fig. 5 show this trend. We observe that 12 sequences represent 70% of mobility for N = 16, while for N = 64 approximately 800 sequences are needed to represent the same percentage.

VI. PERFORMANCE EVALUATION

This section evaluates the performance of the prediction algorithms proposed in Section IV. The goal is to use Algorithm 2 to estimate the future vehicle position, here represented by the next cell. The evaluation methodology is presented in Subsection VI-A. Next we consider the spacial region described in Subsection V-B to characterize the accuracy of the estimation process. The accuracy and the computation time of the algorithms are assessed in Subsection VI-B, always considering the same grid (N = 16), and in Subsection VI-C for different grids (N = 16, N = 64, N = 256).

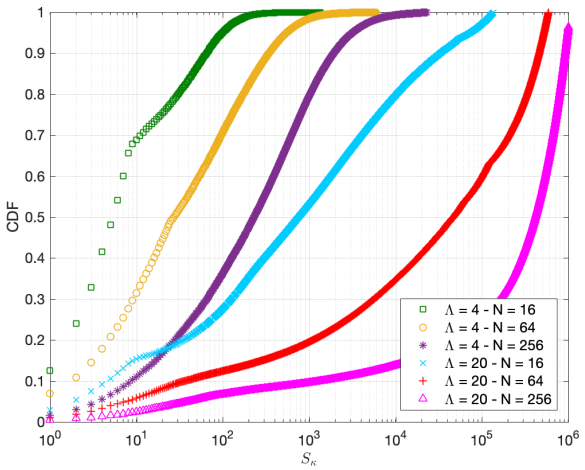

FIGURE 5. CDF of sequences' occurrence in the dataset - N = 16 vs N = 64 vs N = 256.

A. EVALUATION METHODOLOGY

The dataset used in the validation contains the trajectories of 442 taxis running in the city of Porto during a full year, as described in Subsection V-A. We choose a grid map representation with N cells, and we filter the original dataset to consider the trajectories contained in that area (paths starting and ending within the defined area). After characterizing each trajectory as a list of cells, we compute the set of sequences Φ to be used as input of the inference model. The training data set contains Ω sequences, which vary for the different values of Λ, as described in Table 4. Thus, we analyze Φ to consider all transitions between consecutive sequences to populate the transition matrix. We also compute the emission matrix by calculating the probability of each cell belonging to every sequence. By doing so, the Hidden Markov model is defined, and the prediction Algorithms 1 and 2 can then be run.

The validation method used to assess the prediction process is based on the output of Algorithm 2. We evaluate the prediction performance by comparing the predicted sequence S_{pred}, with the correct trajectory sequence, S_{corr}, effectively traveled by a vehicle.

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

$$PP = \frac{1}{|\mathcal{T}^{test}|} \sum_{j=1}^{|\mathcal{T}^{test}|} F(S_{pred_j}, S_{corr_j}), \tag{8}$$

where $F(S_{pred_j}, S_{corr_j})$ is 1 if the predicted sequence S_{pred_j} is equal the true sequence S_{corr_j}, and 0 otherwise. $|\mathcal{T}^{test}|$ denotes the cardinality of the set $\mathcal{T}^{test}$.

In the evaluation of the prediction performance we consider 5 values of Λ, 3 values of N, and 2 test datasets ($\mathcal{T}^{test}$) of sequences. The datasets comprise of:

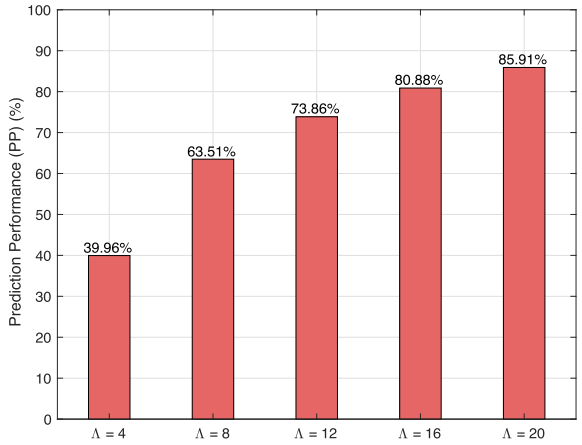

FIGURE 6. Prediction performance for all unique sequences observed in Φ.

- All unique sequences contained in Φ, i.e. $\mathcal{T}^{test} = \{S_1, S_2, \ldots, S_\Psi\}$;
- 10^5 sequences randomly selected from Φ.

Additionally, the computation time is the performance for a dataset composed of 10^3 sequences randomly selected from Φ.

The experiments are carried out in Python through NumPy package and executed on a Intel 8-core i7-9800X processor @ 3.8 GHz and 128 GB of memory.

B. EXPERIMENTAL RESULTS FOR A SINGLE GRID

In this Subsection, N is fixed to 16. First, we evaluate the prediction performance for $\mathcal{T}^{test}$ that includes all unique sequences found in Φ. From the results reported in Fig. 6, we observe that the prediction process is improved with the increase of Λ. The results indicate that better prediction results are achieved when more prior information is considered in the prediction of the next cell.

Instead of considering only the unique sequences, in Fig. 7, we analyze the prediction performance of Algorithm 2 for a test dataset with 10^5 sequences randomly observed in Φ. As can be seen in Fig. 7, the prediction performance increases once again with Λ, but more slowly when compared with Fig. 6. However, when random sequences are selected from Φ, the most probable sequences occur in $\mathcal{T}^{test}$ with a higher probability for $\Lambda = 4$, as shown in Fig. 4. For this reason, the increase of Λ has a marginal effect on the prediction performance in Fig. 7.

Table 6 compares the performance of two existing trajectory prediction models with the proposed model OPTVIT. It is worth mentioning that, although the models in comparison consider different mobility scenarios, the proposed model achieves a higher prediction performance. Besides that, OPTVIT prediction performance is close to the limit of user mobility's potential predictability (93%), as reported by Song *et al.* [28].

The Algorithms 1 and 2 achieve the same prediction performance. However, the computation time performance is significantly different, as we can see in Fig. 8. We evaluate the

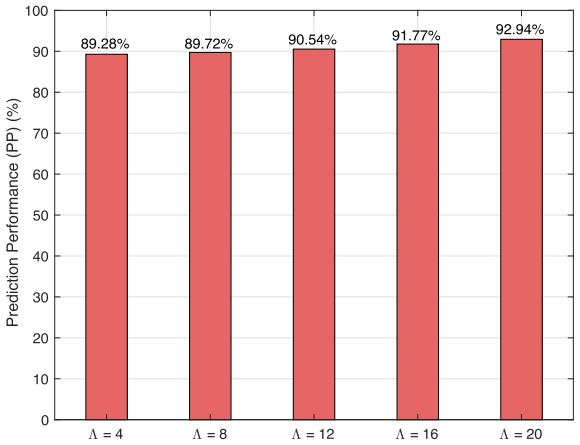

FIGURE 7. Prediction performance for 10^5 randomly sequences observed in Φ.

TABLE 6. Comparison of performance for different existing trajectory prediction models.

Model	Prediction Performance (PP) (%)
HMTP* [1]	80.60%
PPM zone (order 3) [12]	87.35%
Proposed OPTVIT	89.28% – 92.94%

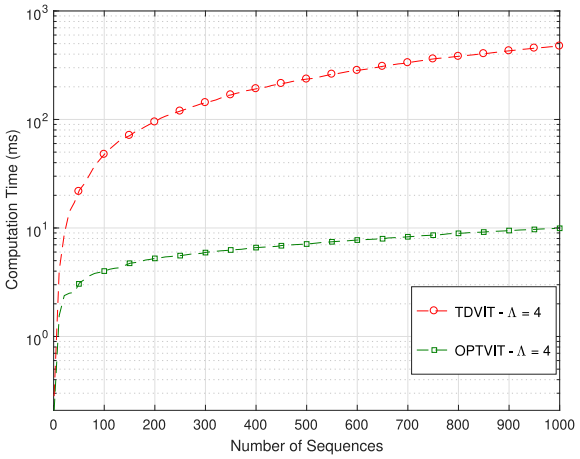

FIGURE 8. Computation time - TDVIT vs OPTVIT.

prediction time between the algorithms (TDVIT vs OPTVIT) for a test dataset with 1000 sequences randomly observed in Φ. The computation time refers to the cumulative time to predict the next location of a number of sequences indicated on the x-axis. The results in Fig. 8 show a significant improvement in terms of computational time when adopting OPTVIT, thus confirming the advantage of the proposed innovations.

Moreover, the computation time of both algorithms increases with the parameter Λ, explained by the increasing number of hidden states and the number of cells that compose the observable state set, as we can see in Fig. 9 for the OPTVIT algorithm.

C. EXPERIMENTAL RESULTS FOR DIFFERENT GRIDS

To assess the influence of the number of the grid cells on the performance of the proposed algorithm, we consider the spatial region described in Subsection V-B that was also

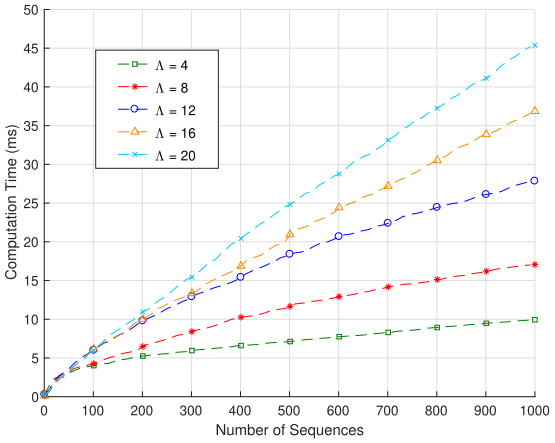

FIGURE 9. Computation time - OPTVIT.

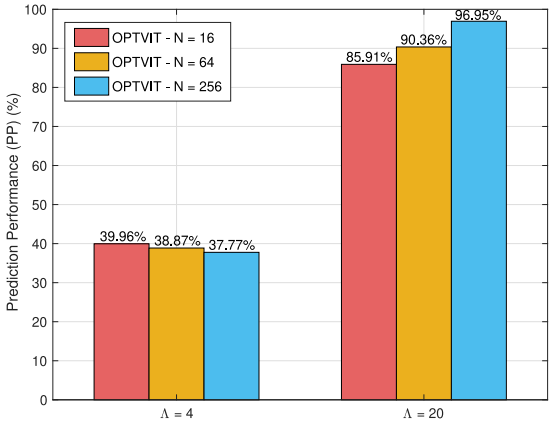

FIGURE 10. Prediction performance for all unique sequences observed in Φ.

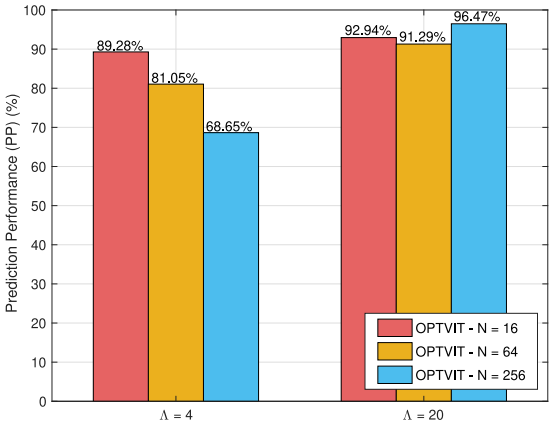

FIGURE 11. Prediction performance for 10^5 randomly sequences observed in Φ.

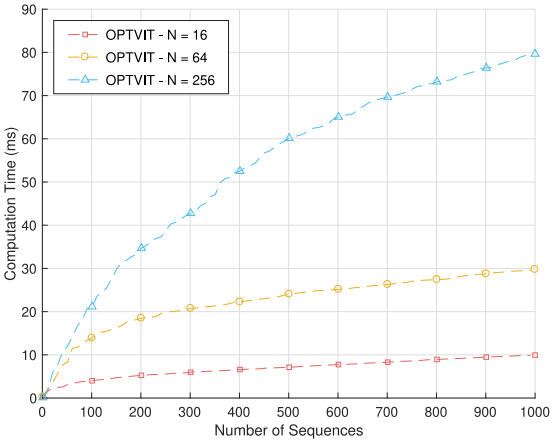

FIGURE 12. Computation time of OPTVIT for $\Lambda = 4$.

considered in Subsection, VI-B and we compare the results for N = 16 (4 × 4 cells) with N = 64 (8 × 8 cells), and N = 256 (16 × 16 cells). We reinforce that the spacial region is always maintained and only the number of cells and its dimensions is changed to cover the same spatial region.

Regarding the prediction performance, we tested OPTVIT for the unique sequences and for 10^5 sequences randomly selected from the mobility dataset. The results are presented in Figs 10 and 11, respectively, considering sequences with 4 cells ($\Lambda = 4$) and with 20 cells ($\Lambda = 20$). For the unique sequences, the results in Fig. 10 show that the prediction performance slightly decreases with N for shorter sequences ($\Lambda = 4$) but increases for longer sequences ($\Lambda = 20$). The decrease for shorter sequences is due to the fact that the number of more dominant sequences increases with N, as shown in the CDF of the sequences presented in Fig. 5, where 12 sequences represent 70% of mobility for N = 16, while for N = 64 approximately 800 sequences are needed to represent the same percentage. For longer sequences ($\Lambda = 20$), the increase of N increases its dissimilarity and quantity, thus describing each mobility pattern in a more detailed manner, which explains the increase of prediction performance from 85.91% to 96.95%.

When 10^5 randomly sequences are considered instead of the unique sequences, the performance decreases substantially with N for shorter sequences but stills improves the performance for longer sequences, as can be seen in Fig. 11

The results in Figs. 10 and 11 indicate that the increase of the grid over the same spatial region can increase or decrease the prediction performance, depending if longer or shorter sequences are considered in the prediction, respectively. This is an interesting result because it shows that the number of cells per sequence is increased by using more prior data, and the prediction performance is improved.

Regarding the prediction computational time, we evaluated the computation time for a test dataset with 1000 sequences randomly selected from the Φ, and considering $\Lambda = 4$. The results are plotted in Fig. 12 for N = 16, N = 64, and N = 256, showing that the increase of more computations leads to a lower computation time per sequence because Algorithm 2 reuses prior computation results stored in $\mathbf{R}$. Moreover, the number of the cells (N) strongly impacts the computation time, because the number of hidden states increases with N. However, we highlight that the computation time per sequence is still below 0.08 ms per sequence (average) in the worst case scenario (N = 256), confirming the practicality of the proposed method. Comparing the average computation times, the algorithm achieves

0.01 ms/sequence for N = 16, 0.03 ms/sequence for N = 64, and 0.08 ms/sequence for N = 256. These results show that the computation time of the algorithm roughly increases by a factor of three when the number of cells is increased by a factor of four.

VII. CONCLUSION

In this paper, we have proposed an accurate and efficient tool to predict the future location of vehicles. We have adopted an HMM that is adequate to model the transitions between sequences (sub-trajectory) and to describe the relation between the cells and the sequences. Having the goal of predicting the next cell a vehicle move to, we first proposed a modified version of the traditional Viterbi algorithm (TDVIT). To reduce the computation time of the prediction algorithm, we also proposed an optimized version of the TDVIT algorithm (OPTVIT). Then, we have designed an evaluation method to assess the prediction process, based on the output of the algorithms, by comparing the predicted sequence, with the correct trajectory sequence followed by each vehicle. The experimental results have shown that the prediction process has improved with the increase of the number of cells per sequence, presenting high accuracy when more observations have been considered before performing the prediction of the next cell. Additionally, we also analyzed the time performance of the two proposed prediction algorithms, concluding that the computation time of TDVIT is higher than OPTVIT and that the computation time of both algorithms grows with the number of sequences and the number of observations.

REFERENCES

[1] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[2] S. Chen, Y. Li, W. Ren, D. Jin, and P. Hui, "Location prediction for large scale urban vehicular mobility," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jul. 2013, pp. 1733–1737.

[3] Q. Lv, Y. Qiao, N. Ansari, J. Liu, and J. Yang, "Big data driven hidden Markov model based individual mobility prediction at points of interest," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5204–5216, Jun. 2017.

[4] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, no. 7075, pp. 462–465, Jan. 2006.

[5] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Jun. 2008.

[6] L. N. Balico, A. A. F. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. B. F. Oliveira, "Localization prediction in vehicular ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2784–2803, 4th Quart., 2018.

[7] A. Ramchandani, C. Fan, and A. Mostafavi, "DeepCOVIDNet: An interpretable deep learning model for predictive surveillance of COVID-19 using heterogeneous features and their interactions," *IEEE Access*, vol. 8, pp. 159915–159930, Aug. 2020.

[8] J. S. Jia, X. Lu, Y. Yuan, G. Xu, J. Jia, and N. A. Christakis, "Population flow drives spatio-temporal distribution of COVID-19 in China," *Nature*, vol. 582, no. 7812, pp. 389–394, Jun. 2020.

[9] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. 10th Int. Conf. Ubiquitous Comput. (UbiComp)*, 2008, pp. 312–321.

[10] J. Ding, H. Liu, L. T. Yang, T. Yao, and W. Zuo, "Multiuser multivariate multiorder Markov-based multimodal user mobility pattern prediction," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4519–4531, May 2020.

[11] S.-H. Fang, L. Lin, Y.-T. Yang, X. Yu, and Z. Xu, "CityTracker: Citywide individual and crowd trajectory analysis using hidden Markov model," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7693–7701, Sep. 2019.

[12] 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. 92850–92860, Jul. 2019.

[13] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101441–101452, 2019.

[14] S. Qiao, N. Han, J. Wang, R.-H. 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.

[15] 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.

[16] 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.

[17] 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, Dec. 2019.

[18] Y. Zhang, J. Hu, J. Dong, Y. Yuan, J. Zhou, and J. Shi, "Location prediction model based on Bayesian network theory," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2009, pp. 1–6.

[19] C.-L. Liu, E. Jou, and C.-H. Lee, "Analysis and prediction of trajectories using Bayesian network," in *Proc. 6th Int. Conf. Natural Comput.*, vol. 7, Aug. 2010, pp. 3808–3812.

[20] 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 (PerCom Workshops)*, Mar. 2015, pp. 469–474.

[21] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, Jun. 2018.

[22] A. de Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio, "Artificial neural networks applied to taxi destination prediction," 2015, *arXiv:1508.00021*. [Online]. Available: http://arxiv.org/abs/1508.00021

[23] (2015). *Taxi Service Trajectory (TST) Prediction Challenge ECML/PKDD*. [Online]. Available: http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html

[24] Y. Tang, N. Cheng, W. Wu, M. Wang, Y. Dai, and X. Shen, "Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3967–3979, Apr. 2019.

[25] F. Altche 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.

[26] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1341–1352, Feb. 2020.

[27] J. Lv, Q. Sun, Q. Li, and L. Moreira-Matias, "Multi-scale and multi-scope convolutional neural networks for destination prediction of trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3184–3195, Aug. 2020.

[28] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

LUIS IRIO received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa (UNL), Portugal, in 2013 and 2019, respectively. From 2014 to 2015, he was a Researcher at CTS-UNINOVA. His Ph.D. research work was developed at the Instituto de Telecomunicações, Lisbon, Portugal, during the period from 2015 to 2019. During the Ph.D. period, he published seven articles in prominent international journals (six of which in the first quarter of the Scimago Journal Rank) and 16 papers in international peer-reviewed conferences. He is currently a Postdoctoral Researcher with the Instituto de Telecomunicações, and his current research interests include mobility prediction, machine learning, and big data analytics.

ANDRÉ IP is currently pursuing the M.Sc. degree in electrical and computer engineering with the Nova University of Lisbon. His research interests include machine learning and mobility modeling.

RODOLFO OLIVEIRA (Senior Member, IEEE) received the Licenciatura degree in electrical engineering from the Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa (UNL), Lisbon, Portugal, in 2000, the M.Sc. degree in electrical and computer engineering from the Instituto Superior Técnico, Technical University of Lisbon, in 2003, and the Ph.D. degree in electrical engineering from UNL, in 2009. From 2007 to 2008, he was a Visiting Researcher with the University of Thessaly. From 2011 to 2012, he was a Visiting Scholar with Carnegie Mellon University. He is currently with the Department of Electrical and Computer Engineering, UNL, and is also affiliated as a Senior Researcher with the Instituto de Telecomunicações, where he researches in the areas of wireless communications, computer networks, and computer science. He serves in the Editorial Board of *Ad Hoc Networks* (Elsevier), IEEE Open Journal of the Communications Society, and IEEE Communications Letters.

MIGUEL LUÍS (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal, in 2009 and 2015, respectively. He is an Adjunct Professor with the Instituto Superior de Engenharia de Lisboa (ISEL) and a Researcher with the Instituto de Telecomunicações (IT), and has been involved in several national and European research projects targeting new communication mechanisms for mobile networks. He is currently the coordinator of project ''InfoCent-IoT: Efficient information centric networks for IoT infrastructures'', and he contributes to several other research projects such as Aveiro STEAM City (EU-UIA program), SNOB-5G (FCT-MIT program) and 5G-Perfecta (Celtic-P2020) to name a few. He has published more than 50 scientific works, including three book chapters and 19 publications in peer-reviewed international journals. His research interests include medium access control for wireless systems, routing and dissemination mechanisms for mobile networks and management, orchestration, and softwarization of future networks.