Systematic Analysis of Fine-Grained Mobility Prediction With On-Device Contextual Data

Huoran Li, Fuqi Lin, Xuan Lu, Chenren Xu, *Member, IEEE*, Gang Huang, *Member, IEEE*, Jun Zhang, Qiaozhu Mei, *Member, IEEE*, and Xuanzhe Liu, *Member, IEEE*

Abstract—User mobility prediction is widely considered by the research community. Many studies have explored various algorithms to predict where a user is likely to visit based on their contexts and trajectories. Most of existing studies focus on specific targets of predictions. While successful cases are often reported, few discussions have been done on what happens if the prediction targets vary: whether coarser locations are easier to be predicted, and whether predicting the immediate next location on the trajectory is easier than predicting the destination. On the other hand, while spatiotemporal tags and content information are commonly used in current prediction tasks, few have utilized the finer grained, on-device user behavioral data, which are supposed to be more informative and indicative of user intentions. In this paper, we conduct a systematic study on the mobility prediction using a large-scale real-world dataset that contains plentiful contextual information. Based on a series of learning models, including a Markov model, two recurrent neural network models, and a multi-modal learning method, we perform extensive experiments to comprehensively investigate the predictability of different types of granularities of targets and the effectiveness of different types of signals. The results provide insightful knowledge on what can be predicted along with how, which sheds light on the real-world mobility prediction from a relatively general perspective.

Index Terms—Mobility prediction, user behavior analysis, multi-modal learning

1 Introduction

Human mobility prediction has drawn increasing attention in the past few years. Predicting the next location of a user is widely expected to be helpful for many applications and services, including but not limited to smart transportation, personalized service recommendation, public resource management, and so on. So far, a large amount of mobility prediction methods have been proposed, ranging from pattern-based methods [1], [2], [3], [4], [5], to Markov model-based methods [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], and to deep neural networks [18], [19], [20], [21]. These methods are applied to various scenarios, including indoor walking [9], venue recommendation [15], urban commuting [19], or even intercontinental trips [18]. Successful stories are often reported, with improved accuracy numbers on particular prediction targets.

Despite these continuously advanced models and improved results, some fundamental questions of mobility prediction have not been well answered, or even discussed systematically.

First, does the granularity of targets matter for the prediction? In practice, the granularity of the next location that a system can predict is critical for the feasibility of real world applications. For example, when a tourist plans a travel to New York in the near future, predicting whether they will go to Manhattan or Central Park makes difference. When recommending a restaurant for the tourist, the granularity of prediction target really matters, which can probably affect the recommendation results and user experiences. Given the same signals (e.g., behavioral data on their smartphones), is it more difficult to predict Central Park compared to Manhattan as the target, or vice versa? Does predicting finer-grained location require a more complex model? Existing studies mainly concentrate on a particular type of targets, mostly due to their task or the data they have access to (e.g., check-in logs), and few have taken into account the impact of the granularity of their prediction targets (locations).

Second, does the salience (or meaningfulness) of next location matter? Almost all existing studies are aimed to predict the *exact* next location that a user is going to access [3], [4], [6], [7], [8], [13], [17], [18], [19], [20], [22], [23]. These efforts are all made at a fixed granularity, and do not distinguish the user intention of the visit. For instance, the proposed prediction models usually care about whether "*the next location on the user's trajectory is a coffee shop*", but usually do not consider whether the coffee shop is just a temporary stop, or the real destination that the user plans to stay. If the user just wants to stay at the coffee shop for a short time (e.g., waiting for their friends), the coffee shop is a temporary stop that may be a less meaningful location. In contrast, the next location where the user and their friends will stay, e.g., the shopping mall or restaurant, should be much more meaningful for the location-based service. If the user would like to have a talk with their

- *Huoran Li, Fuqi Lin, Gang Huang, and Xuanzhe Liu are with the Key Laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing 100871, China. E-mail: {lihuoran, linfuqi, hg, liuxuanzhe}@pku.edu.cn.*
- *Chenren Xu is with the Software-hardware Orchestrated Architecture Laboratory, Peking University, Beijing 100871, China. E-mail: chenren@pku.edu.cn.*
- *Jun Zhang is with the Alibaba Group, Hangzhou, Zhejiang 311121, China. E-mail: zj157077@alibaba-inc.com.*
- *Xuan Lu and Qiaozhu Mei are with the University of Michigan, Ann Arbor, MI 48109 USA. E-mail: {luxuan, qmei}@umich.edu.*

Manuscript received 13 July 2019; revised 3 June 2020; accepted 24 July 2020. Date of publication 11 Aug. 2020; date of current version 3 Feb. 2022. (Corresponding author: Xuanzhe Liu.) Digital Object Identifier no. 10.1109/TMC.2020.3015921

friends in the coffee shop for a long time, the coffee shop can be viewed as a more meaningful destination of their trip. In practice, user's movements are continuous, and predicting the next "meaningful" location is much more useful than predicting the "just next" location. However, how to define the meaningfulness of a location is non-trivial. It can be figured out only when the user's activities at every single location are available. Otherwise, the salience of a location could be judged by how long the user stays. In such a context, a key question needs to be answered, i.e., how difficult is it to predict the next sustainable location, compared to predict just the next location?

Last but not least, how much do different behavioral signals matter in the prediction? Existing studies usually build the prediction models based on various features that correlate to the fixed granularity of locations, such as historical locations along with timestamps, or semantic tags of locations. Indeed, these features are intuitively predictive for future locations, at the same granularity. However, when predicting the next locations at varying granularities, or locations with different intents/duration of stay, none of existing studies have explored whether the features are still effective. On the one hand, in addition to the "location records," there can be many other more meaningful and informative signals from the user's behaviors to reflect the user's intent or trajectory. For example, different types of behaviors and system status can be collected from the user's smartphone, at different location granularities, such as user's app usage behavior and smartphone's sensor data. These fine-grained behavioral signals are usually not covered by existing studies, as collection tasks are non-trivial, time-consuming, and even related to ethic issues. Hence, so far we have no idea of whether these behavioral signals are useful for mobility prediction. In addition, it remains unclear whether such signals contribute to trajectory, especially in the context of varying prediction targets.

We take the initiative to bridge the knowledge gap by addressing the preceding questions. We conduct a systematic analysis of predicting the next location with the user contextual information. We conduct our study based on a recently collected large-scale dataset of contextual usage on smartphones, including various data such as location data, salience data, and behavioral data including app usage data, location sensor data, and broadcast data. Rather than attempting to find the best model for a specific setup (as done in most existing studies), we focus on comparing and analyzing the prediction problem setups under various *granularities* and *salience* (duration of stay) of the target locations, and different types of *behavioral signals* as features, with a set of *prediction models*.

The major contributions of our work are as follows:

- To the best of our knowledge, we make the first systematic study on how the variations of problem setups (with respect to the contextual data) can affect the performance of human mobility prediction. To be more concrete, our paper discusses the impact of the granularity and salience of target locations, as well as different behavioral features on the prediction accuracy.
- We carefully design an empirical experiment to analyze the impact of the preceding contextual factors. Based on a comprehensive, multi-grained, real-world dataset, we conduct a series of study, qualitatively and quantitatively, to address the preceding questions. The results reveal many interesting patterns of user mobility along with useful insights.
- We present design implications derived from our study, which can guide the building of applications of mobility prediction in practice.

The rest parts of this paper is organized as follows. We first introduce related literature in Section 2. Then, we describe the scope of this paper and our analysis pipeline in Section 3. After that, we present the details of dataset in Section 4. Experimental settings and results are presented in Sections 5, 6, and 7. Section 8 presents implications based on the experiment results and discusses limitations. We end our paper with concluding remarks and future outlook in Section 9.

2 Related Work

Human mobility prediction has increasingly drawn attention in the past few years. Researchers have already proposed a variety of prediction models based on various technologies, including pattern-based models [1], [2], [3], Markov-based models [6], [7], [15], and neural network models [18], [19], [20]. The goal of most existing studies is mainly to optimize the model under a rather fixed setting, i.e., a fixed location granularity, a particular target salience, and a specific set of features. For example, Feng *et al.* [19] designed a model based on attentional recurrent networks to improve the performance of mobility prediction. Gao *et al.* [23] leveraged Bayesian techniques and CNN kernel to design VANext model, which outperformed the existing RNN-based models. Both of them focused on optimizing the performance of mobility prediction under only a fixed setting. In contrast, we try to figure out the impact of location granularity, target salience, and behavioral signals on mobility prediction tasks. We can categorize the related literature from location granularity, target salience, and involved features, respectively.

2.1 Location Granularity

In most cases, the location granularity is determined by the prediction task or the data that existing studies can access. Generally, location data have three forms. A "location" is actually a *point of interest* (POI) (e.g., check-in data [5], [8], [12], [13], [15], [18], [19], [20], [22], [24], [25]), a connected region (e.g., a region covered by a base station [2], [10], [11]) or a surveillance camera [6], [17], or a pair of coordinates (e.g., GPS coordinates [1], [3], [4], [7], [9], [16], [21]). For the POI data, the location granularity is fully determined by the granularity of POIs. For the regional data, the location granularity is measured as the average size of all regions. For real-value coordinates, existing studies usually convert continuous data into discrete regions. As a result, the granularity of locations essentially refers to the size of regions. In these studies, once the data is processed, the location granularity usually keeps the same and never changes. For example, Liu *et al.* [18], Feng *et al.* [19], and Gao *et al.* [23] evaluated their models with real-world check-in dataset, i.e., Gowalla Dataset [26] and Foursquare Dataset.[1] The location in such datasets is collected at

1. https://sites.google.com/site/yangdingqi/home/foursquare-dataset

the same granularity, which is represented by the latitude and longitude of the location. However, our work aims to evaluate how the varying location granularity can influence the mobility prediction. Such a task requires a dataset covering various location granularities. To this end, existing datasets are inadequate.

As stated previously, the granularity of the location to be predicted is critical for real-world applications. However, existing studies usually design models based on a fixed granularity without considering the possible impact of the granularity on the performance of prediction. When different datasets with different location granularities are used [18], results are reported for each of the datasets, respectively.

2.2 Target Salience

In practice, not every location is meaningful and worth predicting as the intended target. User's trajectory traces are continuous. Before reaching the intended target, the user can pass by many intermediate points. Predicting the user's real destination in the future is more practically meaningful, rather than those which they only pass by. Hence, we should carefully take into account each location's salience. Intuitively, only a location whose salience is long enough should be considered as the meaningful prediction target.

Unfortunately, to the best of our knowledge, none of all existing studies made in-depth considerations of the concrete salience of each location, nor distinguished which locations are worth predicting with respect to location salience. Consequently, existing studies usually took only the "exact" next location that the user will visit as the intended target. Although some efforts [2] defined three different prediction targets and compared the performance under different targets, the impact of salience is still primitive. Therefore, it is worth exploring the impact of salience.

2.3 Involved Features

Intuitively, features that are most relevant to mobility prediction are users' historical locations and the corresponding timestamps. Almost all existing studies involve these two kinds of information into their prediction models. In addition, in order to better understand the semantic meanings of locations, some studies also involved semantic tags of locations in the models [1], [4], [15], [20], [21], [22]. Ying *et al.* [1], [4] used some general categories of the landmarks as their semantic tags to train their models. Zhang *et al.* [15] and Yao *et al.* [20] used geo-tagged social media data, which include not only spatial and temporal information, but the movement and activities of users as well. Wu *et al.* [21] considered the semantic tags of each road. Cheng *et al.* [22] involved the location's detailed information and social media information. Indeed, users' historical locations, the corresponding timestamps, and the preceding semantic tags of locations are all closely related to user's trajectory. Therefore, they are naturally informative for predicting future locations. However, designing a mobility prediction model based on only these features is far from sufficient and satisfactory. In addition to these "location records", there can be many other signals that possibly indicate user's interests and behaviors on a specific location. These signals, including usage logs and system status, are rather useful, or even more indicative of the user's intent or movement. For example, Nadai *et al.* [27] compared human behavior between the digital world and the physical world. They reported many similarities between app usage and user mobility, which implies that app usage signals can be used in mobility prediction tasks. Plenty of efforts have been made to analyze these signals and applied them to a variety of other studies [28], [29], [30]. Unfortunately, due to the lack of such type of data, existing studies of mobility predictions seldom explored these fine-grained behavioral features. It should be interesting to explore whether these features do contribute value, and how their effectiveness varies against different prediction targets.

3 Problem Statement And Analysis Pipeline

As stated in the introduction, the scope of this paper is to explore the impact of problem setups on prediction accuracy in mobility prediction scenarios. To be more specific, a problem setup consists of three components: location granularity, target salience, and input features. To better clarify the scope of our work, in this section, we present a formal description of the related concepts and the mobility prediction task, introduce the research questions we aim to answer, and describe the pipeline of the analysis.

3.1 Mobility Prediction Formulation

Definition 1 (Location). *A location l is defined as a region of connected area. Each location is identified by a numerical ID.*

Definition 2 (Location Granularity). *Location granularity G is the average area of all locations. We indicate that the locations are fine-grained with a small average area, or coarse-grained with a large average area. l^G denotes a location at the location granularity G. We ignore the superscript when it does not cause ambiguity.*

Definition 3 (Location Record). *A location record r is a tuple of a timestamp t and a location identification l, i.e., $r^G = (t, l^G)$. A location record can tell us where the user is at a specific moment. We also ignore the superscript here when it does not cause ambiguity.*

Definition 4 (Trajectory). *Given a user u and a time window w, a trajectory is a sequence of location records $T_w^u = r_i r_{i+1} \dots r_{i+k}$, which illustrates the user's movement in a period of time.*

Definition 5 (Staying Time & Location Salience). *A location's staying time is defined as the duration that the user stays in this location. Formally, if the user enters a location l_j at time t_1 and leaves at time t_2, the staying time of l_j is $S_{l_j} = t_2 - t_1$. The location salience implies the importance of a location. In this paper, we define that the location salience is positively correlated with the staying time. The longer the staying time is, the higher the salience is regarded. In this way, we can use a location's staying time to represent its salience.*

Definition 6 (Target Location). *The target location l_t is defined as the very* first *location whose salience is high enough in the user's future trajectory. The criterion which determines whether a salience is high enough is denoted as C.*

Definition 7 (Mobility Prediction). *The goal of mobility prediction is to predict l_t based on the user's historical trajectories*

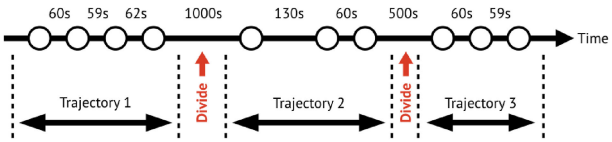


Fig. 1. An example of trajectory extraction. The time intervals pointed by red arrows are larger than 5 minutes (300 seconds), so the entire sequence is divided into three trajectories from these two time intervals, which are pointed by red arrows.

and contextual usage behaviors. Formally, for a specific user u, given a historical time window w_h, a future time window w_f, and a feature set F that corresponds to $T^u_{w_h}$, the goal is to predict l_t that is selected from $T^u_{w_f}$ (according to C) based on F.

3.2 Research Questions

The research questions that we want to answer are as follows:

RQ1: What is the predictability of the mobility prediction task under different location granularities? In other words, we aim to know how the location granularity G can affect the prediction accuracy.

RQ2: What is the impact of target salience on the prediction performance? It refers that we aim to investigate the trends of the prediction accuracy with the varying selection criterion C.

RQ3: What is the significance of multiple behavioral signals? On the one hand, we want to know the predictive power of multiple types of behavioral signals. On the other hand, we also want to identify whether and how these features' effectiveness can vary with respect to different prediction targets.

3.3 Analysis Pipeline

To answer the preceding research questions, we propose an analysis pipeline that can systematically measure the impact of problem setups on prediction performance. The pipeline consists of four steps: 1) Data Pre-processing, 2) Location Granularity Analysis, 3) Target Salience Analysis, and 4) Involved Feature Analysis. The details of every single step are presented as follows.

3.3.1 Step 1: Data Pre-Processing

As the first step of our pipeline, we extract user trajectories and the related usage behavioral data from the raw dataset. The extraction process can be done in various ways. For example, a possible method is to consider location records in a single day as a trajectory. In this paper, we divide different user trajectories by a pre-defined time interval. As is shown in Fig. 1, if the time interval between two adjacent locations is shorter than 5 minutes, they are considered as one trajectory. Otherwise, they are separated into two different trajectories. After the trajectories are extracted, we collect usage data that are associated with each trajectory, such as user behaviors, smartphone's status, and context.

3.3.2 Step 2: Location Granularity Analysis

We conduct a descriptive analysis of the processed data. In this step, we can derive several basic statistics of the data, such as transition probabilities between locations and average staying time at every single location. In addition, to quantitatively verify the feasibility of prediction under each location granularity, we build a group of machine learning models based on the historical trajectories, and then apply the models to predict a user's future movement and examine how the models perform under varying location granularities.

3.3.3 Step 3: Target Salience Analysis

In this step, we define some criteria to select the target location, and then investigate how the salience can affect prediction accuracy. We also apply the preceding machine learning models in Step 2 to perform the predictions, and compare the prediction performance of these models under varying prediction criteria.

3.3.4 Step 4: Involved Feature Analysis

At last, we explore the predictive power of multiple types of behavioral signals. To be more specific, we try various kinds of features combinations, to see which of them can significantly contribute to the mobility prediction task. Since different kinds of features have different forms, we have to adopt a multi-modal learning method to synthesize different features.

4 The Dataset

In this paper, we use the released Sherlock dataset [31], maintained by the BGU Cyber Security Research Center. The Sherlock dataset is a long-term and comprehensive time-series collection of on-device contextual information of smartphones. In this section, we briefly introduce the dataset, and explain how we extract related data from the raw dataset. More details of this dataset can be found on the official website.[2]

4.1 Dataset Overview

The Sherlock dataset contains a multidimensional time-series recording nearly *all software signals and hardware sensors* that can be obtained from a Samsung Galaxy S5 smartphone, without root privileges. The creators of the dataset recruited a group of 50 volunteers, each of whom were assigned a Samsung Galaxy S5 smartphone as the major device in regular life. Each phone was pre-installed with a data collection agent (an Android app). With the consent of the volunteers, their usage data were recorded by the agent. In this paper, we adopt data spanning three months and covering 50 participants.

The agent collects data in two ways: active collection and passive collection. The active collection means that the agent reads information and records it periodically. The passive collection means that the agent makes a record when an event is triggered (for example, when a phone call comes in). Overall, the Sherlock dataset consists of over 600 billion data points in over 10 billion data records, which are sufficient to conduct our analysis.

4.2 Location Data

In this dataset, the exact geo-locations of the volunteers were anonymized by the creators due to the privacy

2. More details of Sherlock dataset can be accessed via http://bigdata.ise.bgu.ac.il/sherlock/

preserving requirement. Instead, the creators performed a K-Means clustering algorithm for all the volunteers' occurrences. Only the cluster IDs of the users' locations are reported in the dataset. These IDs do not have any geographic information. They can be used only as categorical identifiers. In this way, all the user movement range is divided into several locations (clusters).

We have no knowledge about the actual value of the entire region size and the area of locations at each granularity. In other words, the location granularity in this dataset is described with a relative value. A higher number of clusters can indicate a finer granularity.

In the Sherlock dataset, there are six independent settings of M (number of clusters) in the K-Means clustering: 5, 10, 25, 50, 75, and 100. There are no hierarchical relationships between different Ms. For example, the location division under $M = 25$ is not a subdivision of the location division under $M = 50$. The location records are actively collected at a frequency of around once per minute. Each location record contains six IDs, each of which corresponds to the location ID under six different Ms, respectively.

4.3 User Behavioral Data

The Sherlock dataset contains rich information of user behaviors. However, not all of them are suitable for the mobility prediction. Some features are not practically usable because they are too sparse, such as SMS log, call log, and app changing log. Other features, such as screen brightness and speaker volume, are not considered because they are inherently not relevant to user's movement. In this study, we select three groups of behavioral features:

- *App usage data*. App usage data are actively collected. We can know what apps are running (both in foreground and background) at every five-seconds interval.
- *Location sensor data*. Such data can include many sensors that are related to smartphone's motion and gesture, such as accelerometer, gyroscope, orientation, and barometer. They are actively recorded for every 15 seconds. Our dataset consists of 238 sensors in total. More details of sensors can be accessed via an external link.[3]
- *Broadcast data*. Whenever an Android system broadcast is triggered, its content will be passively recorded by the agent. There are 82 kinds of broadcasts.

App usage data can indicate the users' active usage behavior and intent. The other two kinds of features can represent the device's system-level status. In particular, they can also cover geo-location related system status (e.g., speed and acceleration of the smartphone). Therefore, we choose the above three groups of behavioral features, and we believe these features can provide an informative and representative description of users' usage from multiple aspects.

4.4 Trajectory Extraction

We examine and construct usable trajectories from the location records. Certainly, we can arbitrarily intercept any part from a user's location sequence as a trajectory record. However, it is not a practical strategy because there can be missing records. Such a case happens either when the device was powered off, or when the agent failed to record data. In either case, the time interval between two consecutive records could be much longer than one minute. As a result, we do not have enough information about the user movement during this absent time period, so that we have to discard data. In this study, we set a threshold as five minutes, i.e., each pair of consecutive location records that are smaller than five minutes is put in the same trajectory. Fig. 1 presents an example of the trajectory extraction. Since the sampling interval is about one minute, the number of records in a trajectory can approximately equal to the duration of the trajectory (in minute).

To make meaningful predictions, we filter out trajectories that are too short. In this paper, this threshold is set to be one hour, as we think that trajectories shorter than an hour can not perform sufficient information to understand the user's movement. Finally, we obtained *4,785* independent trajectories. The extracted trajectories keep the original form of the raw data (i.e., actively sampled records).

TABLE 1
The Average Staying Time ($Avg.Stay$, in minutes) in a Location Under Different Location Granularities

$M>$	5	10	25	50	75	100
$Avg.Stay$	161.22	75.62	10.02	3.99	2.76	2.42

5 Location Granularity Analysis

We then conduct our study over the data extracted from the Sherlock dataset. As the first part of the analysis, we discuss the impact of location granularity on the mobility prediction. We begin with a descriptive analysis on the location granularity to check some basic characteristics of the data. Then, we make a quantitative analysis to rigorously compare the prediction performance under different location granularities.

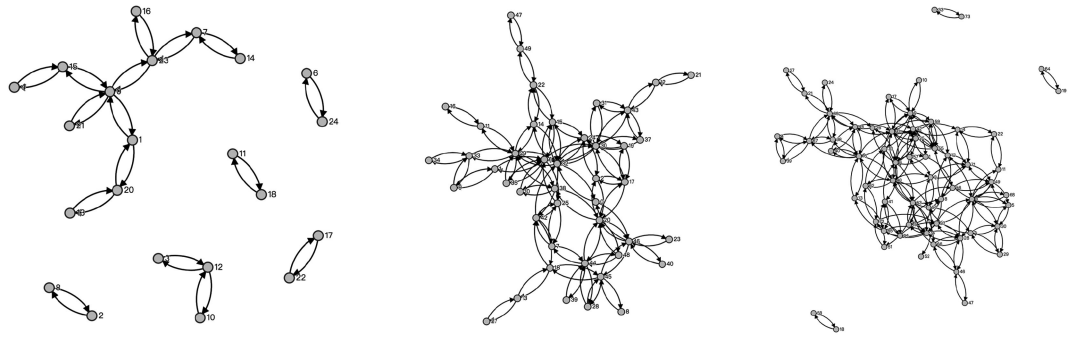
5.1 The Descriptive Analysis

As stated previously, the location granularity in the Sherlock dataset is a relative value. In other words, we do not know the actual value of locations at each granularity. In order to derive a further understanding of each granularity, we characterize the location data.

We first calculate the average staying time in a location (Definition 1) under different location granularities (listed in Table 1). First of all, we can see that the average staying time decreases rapidly with the increment of M (Definition 2), which is as expected. In practice, a user is unlikely to switch their position too sporadically or too frequently. In other words, the average staying time should not be too short or too long. In this sense, a "location" under $M = 25$ or $M = 50$ is more likely to be reasonable.

We then explore the transitions between different locations under varying location granularities. For each pair of locations $< l_i, l_j >$ (Definition 3), we compute the number of transitions from l_i to l_j, then construct a transition matrix with the shape $M * M$. After that, we discard transitions that occurred too few ($<$ 1,000 occurrences) and use the

3. List of sensors on Sherlock can be found at https://drive.google.com/file/d/0B_A1qX1kf7R9Q0llRWpkY2pXdzg/view



(a) $M = 25$ (b) $M = 50$ (c) $M = 75$

Fig. 2. The transition diagrams under three location granularities.

remained transitions to construct a directed acyclic graph (DAG). Due to page limit, we show the DAG under $M = 25$, 50, and 75, respectively, in Fig. 2. When $M = 25$, we can observe various separated clusters. When $M = 50$, all the locations are connected, but the structure of the transitions is still clear. When $M = 75$, however, the transition diagram becomes too dense, even a little bit messy. In summary, from the transition aspect, we can confirm the previous conclusion, i.e., $M = 25$ or $M = 50$ are more reasonable to judge the user mobility in our study.

5.2 Quantitative Analysis

The preceding descriptive analysis provides some intuitive observations about each location granularity. We then conduct a quantitative experiment to rigorously examine the prediction under varying location granularities. In a nutshell, we simulate a group of prediction queries over the user trajectories, train a prediction model, and analyze the accuracy under each location granularity.

5.2.1 Query Simulation

To simulate a location prediction query, we randomly select a time point from the trajectory (Definition 4), and then divide the trajectory into two parts, before and after this time point, respectively: the first part is considered as the historical trajectory of the user ($T^u_{w_h}$), and the second part is the future trajectory of the user ($T^u_{w_f}$) that contains the target location l_t to be predicted. Such a pair $< T^u_{w_h}, T^u_{w_f} >$ can represent a query that occurs at the division point. In practice, suppose that there are n location records in a trajectory. We randomly choose a positive integer m ($m < n$). Then we regard the first m records as $T^u_{w_h}$ and the rest of the records as $T^u_{w_f}$. The location prediction request occurs between $T^u_{w_h}$ and $T^u_{w_f}$, so the last location in $T^u_{w_h}$ can be seen as the user's current location. In order to avoid creating a pair that is unbalanced, we restrict that both $T^u_{w_h}$ and $T^u_{w_f}$ contain at least 20 percent of the original trajectory records. Moreover, we perform five individual simulations on one trajectory to augment the test. Finally, we get 23,925 simulated queries from the 4,785 trajectories.

The experiment follows the standard machine learning pipeline. We split all the 23,925 queries into three groups: a training set, a validation set, and a testing set. To keep the independence of these sets, we preserve that the five queries generated from the same trajectories are placed into the same set. Then, for every single simulated query, we extract the next successive location of the user, from query's $T^u_{w_f}$ as the target location l_t. The next successive location of the user is determined by the first location in $T^u_{w_f}$ that is not equal to the current location (Definition 7). For example, if the current location is A, and the location list in $T^u_{w_f}$ is (A, A, B, B, C, A), the next successive location of this query is B. However, such a definition may not find a next successive location in some queries, i.e., the user never changes their location within w_f. In this case, these queries should not be predicted. We discard these queries during this experiment. The sizes of the testing sets under varying location granularities are listed in the first row of Table 2.

We then employ some typical time-series models to examine the prediction.

5.2.2 The Markov Model

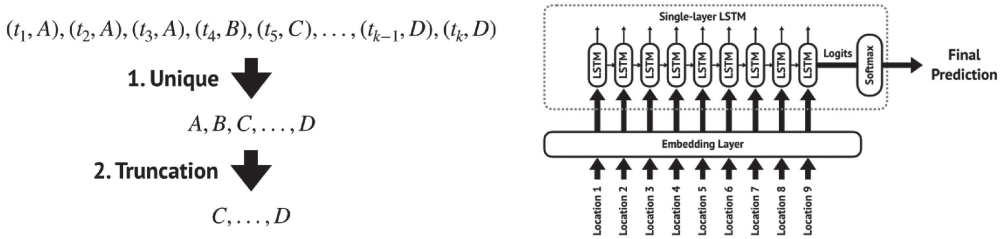
The first employed model is the simple first-order Markov Model. We train the model based on all the queries' historical trajectories, and then use the model to decide which location has the highest possibility to be visited with respect to the current location. Notice that the next successive location should not be the current location, so we remove transitions that a location directly connects to itself. In other words, the diagonal of the transition matrix of Markov model contains only zero.

5.2.3 The RNN Model AND LSTM Model

Building a first-order Markov model is rather straightforward. However, such a model is too naive to capture the complex sequential information from the historical trajectories. As

TABLE 2
Sizes of the Testing Set Under Each Location Granularity and Each Target Selection Criterion

Target Selection Criterion	# of Locations (M)					
	5	10	25	50	75	100
Successive	849	1,429	2,425	2,968	3,078	3,182
Important@2	824	1,368	2,242	2,813	2,945	3,050
Important@5	796	1,276	1,973	2,406	2,476	2,551
Important@10	739	1,180	1,711	1,917	1,864	1,919
Longest@3	208	740	1,935	2,640	2,811	2,933
Longest@5	93	354	1,596	2,392	2,631	2,786
Longest@10	37	146	1,132	1,993	2,284	2,447



(a) Trajectory pre-processing procedure. (b) The LSTM model.

Fig. 3. The trajectory pre-processing procedure and the structure of the LSTM model and vanilla RNN model.

discussed in Section 2, there have been some RNN-based models for mobility prediction [18], [19], [20]. Following the same principle, we employ a vanilla RNN model and an LSTM model over the historical trajectories and predict the next successive location. As is known, LSTM is a variant of RNN, and can better capture long-term dependencies. Compared to the vanilla RNN, LSTM uses LSTM unit that consists of a cell, an input gate, an output gate, and a forget gate, to mitigate the vanishing gradient problem. As the location ID is a categorical value, we actually implement a vanilla RNN classifier and an LSTM classifier.

We perform a two-stage processing of each trajectory to make it more adequate for the RNN and LSTM model. As demonstrated in Fig. 3a, for each trajectory, we merge the repeated consecutive locations into one (the "Unique" step). This step indicates that we consider where the user has visited. We use only the latest 100 locations (the "Truncation" step). The structures of RNN and LSTM model are illustrated in Fig. 3b. We take every single location record as a time step, while each location is represented by an embedding layer before fed into the learning model. The output of the last layer is a logit, a M-dimensional vector, and a Softmax layer is applied to produce the final prediction.

We implement a single-layer RNN model and a single-layer LSTM model with $hidden\_size = 256$ based on the TensorFlow.[4] We adopt *cross-entropy* as the loss function and the `tf.train.GradientDescentOptimizer` as our optimizer. When training the model, we adopt default values, i.e., we set the *learning rate* as 0.05 for the RNN model, and set the *forget bias* as 0.03 and the *learning rate* as 0.05 for the LSTM model.

5.2.4 Experiment Results

The experiment results are shown in Fig. 4. We use Accuracy@1 to evaluate the model performance. As for the Markov model (green bars), when $M = 5$ or $M = 10$, the results are quite close, which are 0.254 and 0.259, respectively. Besides, the prediction accuracy decreases when M increases, from 0.210 to 0.107. Intuitively, this result is not surprising, as the user movement is more difficult to be predicted if the location is defined in a more fine-grained location. Through the performance of the random guess (the yellow curve in Fig. 4), we can observe this phenomenon quite clearly. Therefore, for the Markov model, we can conclude that the prediction accuracy is quite low in more fine-grained locations.

As for the RNN model (blue bars) and the LSTM model (red bars), however, we find a different result. On the one hand, the LSTM model and the RNN model significantly outperform the Markov model for all location granularities. For the LSTM model, the results under varying location granularities are 0.495, 0.525, 0.546, 0.430, 0.338, and 0.295, respectively. In contrast, for the RNN model, the results under varying location granularities are 0.486, 0.499, 0.538, 0.400, 0.277, and 0.249, respectively. We can observe that under most location granularities, the accuracy of the LSTM model and the RNN model is more than twice better than that of Markov model. This conveys that the sequential information contained in the historical trajectories can be leveraged much better by the LSTM model and the RNN model.

In addition, we can observe that the accuracy of the LSTM model is a bit higher than the accuracy of the RNN model for all location granularities. Since the LSTM model can better utilize the long-term context compared to the vanilla RNN model, we can infer that the long-term sequential information can be beneficial to improve mobility prediction tasks. On the other hand, the prediction accuracy does not monotonically decrease with the growth of M. The accuracy increases when $M \leq 25$, and then begins to decrease for both models. The best performance is obtained under $M = 25$, which is 0.546 for the LSTM model and 0.538 for the RNN model, respectively.

5.3 Summary

From the descriptive analysis, we can have an intuitive understanding of the impact of location granularity. The location granularity does have significant impacts on the prediction accuracy, and the prediction performance peaks when $M = 25$, indicating that the prediction model outputs with the best

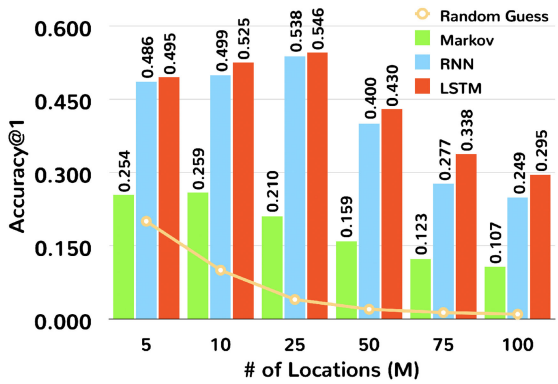


Fig. 4. Accuracy@1 of the next successive location under different location granularities. The green, blue, and red bars represent the Accuracy@1 of the Markov model, the RNN model, and the LSTM model, respectively. We use the random guess as the baseline under each M, which is represented by the yellow curve (i.e., Accuracy@1=$\frac{1}{M}$).

4. https://tensorflow.org

results when the granularity is moderate. The prediction accuracy of the next successive location is reduced if the locations are too coarse-grained or too fine-grained.

6 Target Salience Analysis

In this section, we focus on investigating the impact of target salience (Definition 5) on prediction performance. We define three different metrics to select the target location (Definition 6) based on location salience (i.e., staying time), and explore the prediction performance accordingly. The detailed definitions are:

1) *Successive.* We have already introduced this metric in the previous part of this paper. It takes the first location in $T^u_{w_f}$ that is not equal to the current location as the prediction target. This metric is the most common one that is adopted by almost all the existing studies.
2) *Important@K.* As mentioned above, we think that a location with sufficient staying time is more likely to be an important and meaningful location rather than just a pass-by point. Following this principle, the metric *Important@K is defined as the first location in $T^u_{w_f}$ that the user stays for at least K minutes*. Apparently, the threshold K determines whether a location is regarded to be important. Still, we also require that this location cannot be equal to the user's current location.
3) *Longest@K.* Defining an important location by manually setting a threshold may not be always reasonable, because users usually have very various movement patterns. Thus, a unified "one-size-fits-all" threshold is not suitable. We define the metric Longest@K, denoting *the location with the longest staying time among the first K locations of $T^u_{w_f}$*. We still require that this location cannot be equal to the current location of the user.

We choose $K = 2, 5, 10$ for Important@K and $K = 3, 5, 10$ for Longest@K. Similar to the Successive, the number of labeled queries under different target selection metrics are varied. The testing sizes of all these combinations can be found in Table 2. For Important@K, if there is no location that the user stays for at least K minutes in $T^u_{w_f}$, no label is applied for this query. For Longest@K, if the total number of locations in $T^u_{w_f}$ is less than K, we cannot have enough candidates or select a valid prediction target. Similar to the experiments in the previous section, we conduct the experiments under varying target selection metrics with a Markov model, a RNN model, and an LSTM model, respectively.

6.1 Experiment Results

The prediction accuracy of three models under $M = 25$ is shown in Fig. 5. For clarity, we show the results under only such a location granularity. The performance of the three models present the same trend. For simplicity, we discuss only the results of the LSTM model (red bars) in detail.

We first focus on the performance of Important@K. The accuracy is lower with a larger threshold K. When the threshold of the important location is two minutes, the prediction accuracy is 0.547. When the threshold is 5 minutes, the accuracy degrades to 0.463. When the threshold is 10 minutes, the accuracy keeps degrading to 0.390. The performance of the Markov model and the RNN model present the same trend.

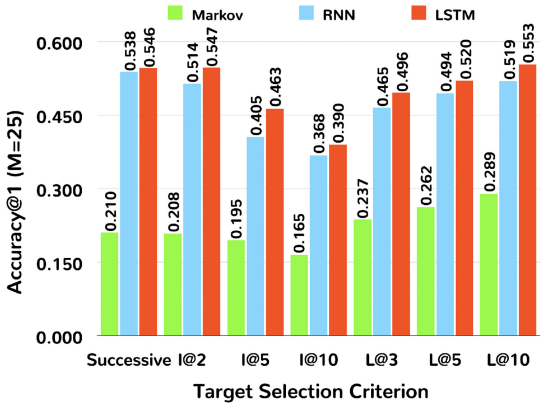


Fig. 5. Prediction accuracy under all target selection criteria. We show only the results for $M = 25$. To save the space, we use "I" to denote Important and "L" to denote Longest.

The accuracy of Markov model degrades from 0.208 to 0.165, while the accuracy of the RNN model degrades from 0.514 to 0.368, respectively. The results indicate that, if the standard for an important location is higher, it is more difficult to reach an accurate prediction. In contrast, when we consider the performance of Longest@K, we can obtain an opposite result. When the threshold of the location with the longest staying time is 3 locations, the accuracy of the LSTM model is 0.496. When the threshold is 5 locations, the accuracy increases to 0.520. When the threshold is 10 locations, the accuracy continues to increase to 0.553. The performance of the Markov model and the RNN model present the same trend. The accuracy of the Markov model increases from 0.237 to 0.289, and the accuracy of the RNN model increases from 0.465 to 0.519, respectively. The results indicate that, if the standard for Longest@K is higher, it is easier to make an accurate prediction.

The above results lead us to the conclusion: when we consider an important location as the target location, the longer the staying time of the target location is, the more difficult the location can be accurately predicted. However, when we consider the location with the longest staying time among the first K locations of $T^u_{w_f}$ as the target location, we can obtain a contrary result. The longest staying time of the first K locations in $T^u_{w_f}$ can always increase with K increasing. Given $M = 25$, the average staying time of the target location under Longest@3, Longest@5, and Longest@10, are 22.3 minutes, 36.3 minutes, and 53.1 minutes, respectively. From Fig. 5, we can observe that the prediction accuracy of Longest@K increases along with the growth of K. Hence, the target location with a longer staying time is much easier to be predicted. Hence, we can find that *the predictability is not simply correlated to the length of the staying time of the target location*. It is more likely to be affected by the form of the target selection metric.

Based on the results, we argue that, *although Important@K is an important metric, Longest@K should also take high importance*.

7 Behavioral Feature Analysis

The preceding sections indicate the impact of location granularity and location salience by the staying time. In this section, we study the prediction power of multiple types of behavioral features. We describe how to extract behavioral features, introduce how to merge usage features into the

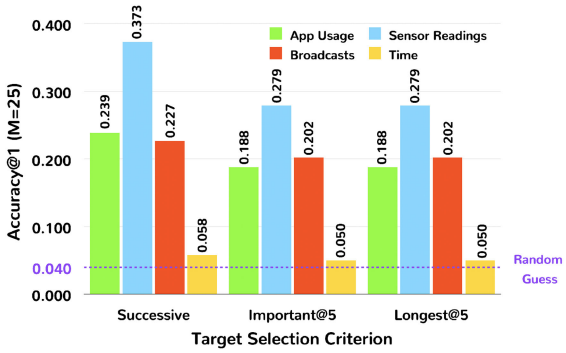


Fig. 6. Prediction accuracy of each single group of features under Successive, Important@5, and Longest@5 under M=25. The red dotted line represents the accuracy of random guess.

RNN model and the LSTM model, and finally analyze the experiment results.

7.1 Involved Features

As stated in Section 4.3, we use the app usage data, the location sensor data, and the broadcasts to illustrate the user's behavior. In addition, we use the time information of a query to describe the temporal context of the user's usage. For each query, we calculate several statistics about the four groups of features within w_h (the time window that corresponds to $T^u_{w_h}$). We list the feature groups as follows:

1) *App Usage.* We count which apps have been used at least once (either in foreground or background) within w_h. Since there are 655 apps in our raw dataset, we use a 655-dimensional binary vector to organize each data entry.
2) *Location Sensor Readings.* We calculate the average value of each sensor's reading within w_h. There are 238 sensor readings in this group of data, so this part of features is represented by a 238-dimensional real-value vector.
3) *Broadcasts.* We use an 82-dimensional vector to represent the broadcast information. Each dimension of the vector records the number of times that a broadcast occurrs within w_h.
4) *Time.* For the temporal context, we roughly use two values: the beginning time and the ending time of w_h. For each of the value, we infer which hour of the day and which day of the week that it belongs to.

7.2 Performance of Every Single Feature

To examine the significance of features, we first try to predict the user's movement purely based on each single group of features. During this step, we adopt a Random Forest Classifier [32] as the prediction model.[5] We denote the combination of a location granularity G and a target selection metric C as a *scenario*. The results of three selected scenarios (Successive, Important@5, and Longest@5) under $M = 25$ are shown in Fig. 6.

The relative relationships among all groups of features are similar under all scenarios. As is shown in Fig. 6, under all scenarios, sensor readings outperform other features with results of 0.373, 0.279, and 0.279, respectively. For the app usage feature, the results are 0.239, 0.188, and 0.188, respectively. For the broadcasts feature, the results are 0.227, 0.202, and 0.202, respectively. For the time feature, the results are 0.058, 0.050, and 0.050, respectively, which are a little bit higher than those of random guess.

In a sense, all of these three groups of features can significantly outperform the random guess. We can observe that models with features of sensor readings can reach the best results, while those with the app usage and broadcasts can get similar performance. We can also learn that location sensors' readings are highly correlated to users' incoming movement. A simple example is that, if we know that the user is moving at a high speed by observing the accelerometer, we can guess that the user may be driving to a faraway location. For the app usage and broadcasts features, we claim that they do provide useful information for the mobility prediction, as they can reflect users' previous usage behavior, which is relevant to the user's incoming movement. Compared to the location sensors, such information does not have quite straightforward and close correlations with the user's movement, and they can not contribute as much as location sensors do.

Considering that users usually have a fixed daily routine, it is a reasonable hypothesis that the features of time are important to users' incoming movement. However, although models with time features outperform the random guess algorithm, the contribution of the time features is not significant in comparison to other feature groups. Such a result indicates that the movements of mobile users at a fixed time are still quite uncertain.

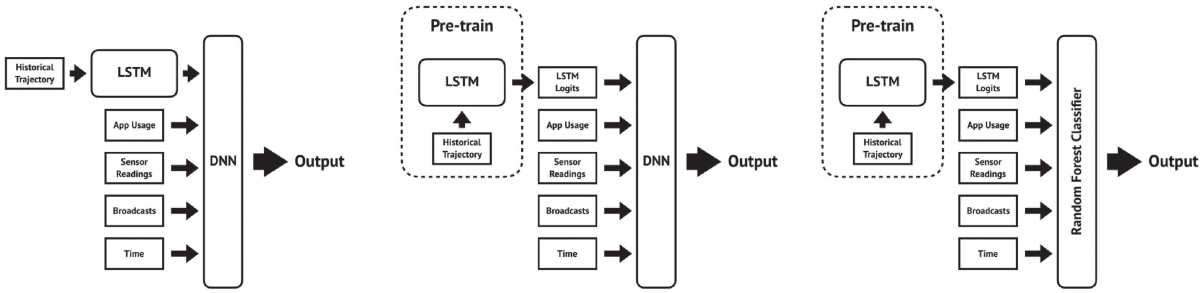
7.3 The Multi-Modal Learning

Although the individual behavioral features are demonstrated to be effective, none of them can produce a better performance compared to the trajectory-based LSTM/RNN model. Indeed, the user's movement should be the major factor for the mobility prediction, while other features may provide auxiliary information. Therefore, it implies that we need to find a way to synthesize these behavioral features into the LSTM model and the RNN model in order to achieve a better result.

The most straightforward way is using the Deep Neural Network (DNN) model to combine all these features. As demonstrated in Fig. 7a, the historical trajectory is first encoded by the LSTM model or the RNN model before it is fed into the DNN. However, this model has similar performance compared to the trajectory-based LSTM and the trajectory-based RNN proposed previously, indicating that it cannot sufficiently make use of the information in the usage features. The reason can be that the DNN is dominated by the strongest signal (historical trajectory), and could neglect other relatively weak features.

We then try to pre-train an LSTM model or a vanilla RNN-based model over the historical trajectory in advance, and use the logits of that LSTM and RNN as a new group of features (Fig. 7b). Unfortunately, this model does not improve much. The problem is due to the fact that DNN cannot make good use of the information from the weak signals. Eventually, we replace the DNN with the Random Forest Classifier (Fig. 7c). This model finally generates results that are better than those from the LSTM model and the RNN model.

5. We use Scikit-Learn package[33] to implement the Random Forest Classifier. All parameters are set to default values.



(a) The end-to-end neural network (b) The two-stage neural network (c) Pre-trained neural network + Random Forest

Fig. 7. The three ways that we tried to integrate usage features into the vanilla RNN model and the LSTM model.

7.3.1 Macro Results

We combine each group of behavioral features with the LSTM logits and RNN logits to train the classifier, then use all the four groups together with the LSTM logits and the RNN logits to train a more comprehensive model.

The results of the mentioned three scenarios (Successive, Important@5, and Longest@5) for the LSTM model are shown in Fig. 8, and the results for the RNN model are shown in Fig. 9. First of all, we can see that each group of behavioral features can make contribution to the mobility prediction for both models. The prediction accuracy can always be enhanced by involving a group of behavioral features. When comparing the significance from the four groups of feature, it is interesting to notice that app usage and sensor readings are the most effective groups when used alone for both models. The differences of accuracy between app usage and sensor reading are relatively low under all scenarios for both models. Although the sensor reading is the most effective group when used alone, it is not always the best when combined with the LSTM model and the RNN model. For the LSTM model, the results of sensor reading are 0.626, 0.502, and 0.556, respectively, which are lower than those of app usage under most scenarios. For the RNN model, although the accuracy of sensor reading is the highest among all groups of features, the difference of accuracy between sensor reading and app usage is rather low, i.e., less than 0.03. This is because the information contained by location sensors and the historical trajectory are highly overlapped. The user's movement status described by location sensors are mostly contained in the historical trajectories. Therefore, integrating location sensors into the historical trajectories cannot provide more valuable information.

Different from this, although the app usage contains less information than location sensors, the knowledge that it contains cannot be contained by the historical trajectories. In other words, although the app usage provides less information, it still has uniqueness that cannot be covered by the historical trajectories. Therefore, leveraging the app usage into the LSTM model and the RNN model can help fill the gap of the accuracy or even make a better performance, compared to synthesizing sensor readings into the same models.

If we look at the performance when using all groups of features, the results for the LSTM model are 0.625, 0.506, and 0.543, respectively, while the results for the RNN model are 0.639, 0.510, and 0.567, respectively. We can observe that *the prediction accuracy is almost the same with only app usage or sensor reading features integrated. Similarly, the contributions of features in the broadcasts and time groups are quite limited.* In summary, we can learn that app usage is the most important group of usage features.

7.3.2 Micro Results

In the preceding analysis, we have shown that user behavioral features can contribute to the mobility prediction. It motivates us to explore the significance of these features under more scenarios. Note that the performance of the pure LSTM model and the pure RNN model under different scenarios can vary a lot. Hence, to make a fair comparison, we define *relative*

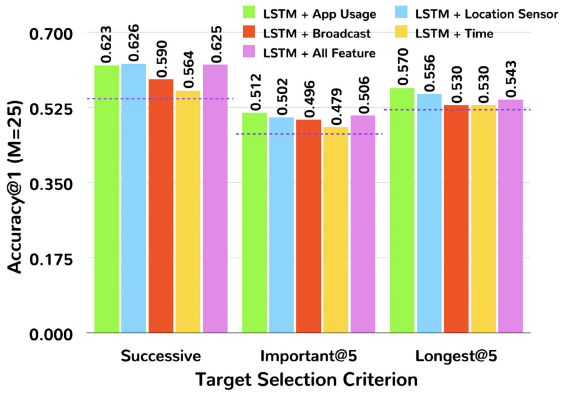


Fig. 8. Prediction accuracy of each combination of features under Successive, Important@5, and Longest@5 under M=25. The dark purple lines represent the accuracy of the trajectory-based LSTM model.

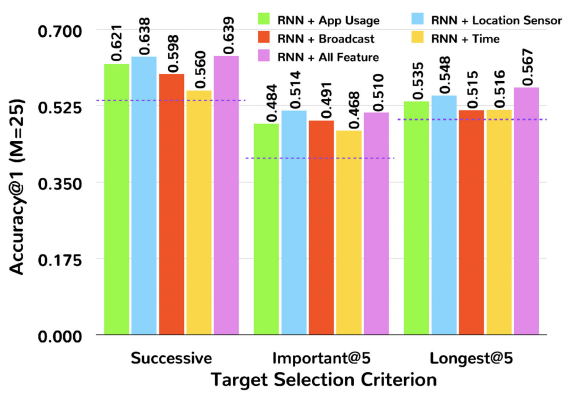


Fig. 9. Prediction accuracy of each combination of features under Successive, Important@5, and Longest@5 under M=25. The black lines represent the accuracy of the trajectory-based RNN model.

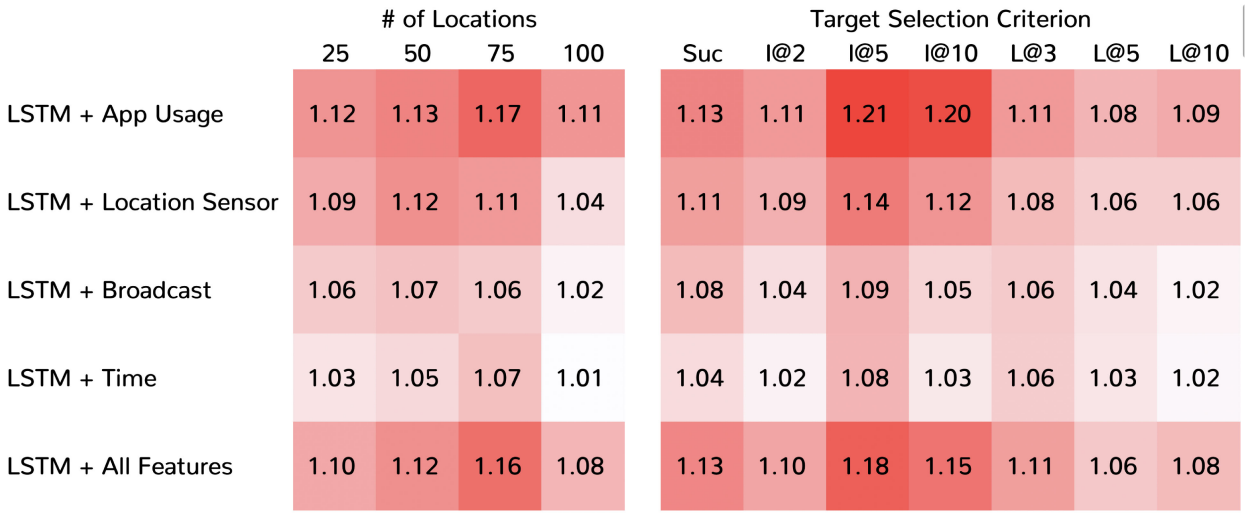


Fig. 10. The average relative performance of usage features under different location granularities and different target selection criteria for LSTM model.

Fig. 11. The average relative performance of usage features under different location granularities and different target selection criteria for RNN model.

performance as the ratio between the accuracy of the synthesized model with the accuracy of the pure deep learning model (pure LSTM model and pure RNN model), and use the relative performance to indicate the significance of the behavioral features under different scenarios.

Since it is hard to visualize a three-dimensional result tensor, we illustrate the average relative performance under each location granularity and each prediction target for the LSTM model in Fig. 10 and that for the RNN model in Fig. 11. Intuitively, the darker the block's color is, the larger the value it holds. When calculating these average values, we do not include results under $M = 5$ or $M = 10$ since the numbers of valid data are too small (see Table 2).

From Figs. 10 and 11, we can obtain some observations. First, the relative relationship among different features remains basically unchanged. The app usage is always the most effective feature, followed by the location sensor. Broadcast and time can make only small contribution. Second, if we focus on the left part of the figure, we can see that the effect of the app usage gets better as the number of locations increases, but drops if the number of locations is 100. Therefore, we can conclude that the app usage is most significant when the number of locations is moderate. In contrast, the effect of other features under different number of locations is not so different. Finally, from the right part of Figs. 10 and 11, we can learn that the behavioral features can make more contributions under Important@K than Longest@K. This means the next important location is more relevant to the user's usage.

8 Discussions

In this section, we first summarize the key findings presented by the above analyses, and propose related implications for future research and practice. After that, we discuss some limitations of our study.

8.1 Summary of Findings and Implications

First of all, *location granularity and salience do have impacts on mobility prediction*. Such impacts have *never* been reported

by existing literature, but deserve more attention in practice. For location granularity, when the location granularity is moderate, a "location" is more likely to be consistent with our common-sense knowledge, and the prediction can be more practical. In addition, the prediction accuracy reaches the peak when the location granularity is moderate, too. This conveys that *when designing a mobility prediction model, the developer should carefully consider the proper location granularities according to their scenario*. It is less likely to generate a usable prediction model with respect to an improper location granularity. For location salience, we can see that the prediction performance varies under different target selection metrics. When taking into account the location salience, the performance decreases compared to predicting the exact next location. Therefore, although existing models can predict the exact next location at a high accuracy, they are not able to efficiently predict the next salient location. Moreover, from our experiments, we find that the prediction performance is not simply correlated to the location salience, but more likely to be affected by selection metrics. In our experiments, *Longest@K is potentially a better selection metric compared to Important@K*, as we can locate the real destination with a higher accuracy in prediction.

Various Behavioral Features are Contextually Helpful. We have demonstrated that behavioral features are quite indicative for the mobility prediction task. When used individually, we find that location sensors are the most predictive behavioral features, because they are more relevant to the user's movement. However, if synthesized with historical trajectories, app usage becomes the most helpful part of features, because it can offer additional information that the historical trajectories do not contain. Therefore, if a developer (e.g., the Location-Based Service provider) tries to build a mobility prediction model that relies on only behavioral data, *they should consider location-related features (e.g., location sensors) as their first choice*. In contrast, if the developer wants to utilize both behavioral features and historical trajectories, *they should pay more attention to features that are more close to the user's active usage (e.g., app usage)*. With the preceding findings and other technologies [34], developers may provide better service to users.

8.2 Limitations

Our current study relies on the Sherlock dataset, which consists of 50 volunteers over a few years. Indeed, such a dataset may be a bit "small" in terms of user scale, and the potential affect should be discussed.

The major goal of this study is to derive some insights on how to optimize the mobility prediction. To this end, we need to build models over various contextual information, including granularities, salience, and some detailed contextual information like app usage, time, and so on. In practice, collecting such data is quite challenging, which is not only due to technical issues, but also requires ethical permissions. As described in Section 4, the Sherlock creators made a lot of efforts to collect and make available this longitudinal and high-dimensional dataset, spanning nearly every single kind of software and hardware sensor on the smartphone. In addition, although only 50 volunteers are involved, the Sherlock dataset consists of over 600 billion data points in over 10 billion data records. In this way, the comprehensive granularity, salience, behavioral information, and time length of Sherlock dataset can ensure the quality and representativeness of the results. Hence, the "depth" of Sherlock dataset makes it the most adequate to date for our study.

We should also mention that the methods and models are not specific to Sherlock. We will release all the code and dataset publicly on the Internet along with the work's publication. Researchers who own a larger dataset containing the similar contextual information can reproduce all the experiments to check the validity and explore more insights based on our contributions.

9 Conclusion and Future Work

In this paper, we rethink the mobility prediction from a new perspective. We have carefully examined how the problem setup can influence the prediction performance. We have designed a general analysis pipeline and conducted a comprehensive study based on a large-scale real-world dataset. We have found some interesting results, which had not been reported by previous studies, and finally brought up several implications.

For future work, we plan to explore more practical scenarios based on the findings of this paper. For example, how to deal with the situation where a user travels to a city for the first time? A possible method is to find out the most similar user (e.g., via social networking services), and leverage their features. Once we have collected sufficient information, we can apply to the new user's own features, e.g., by transfer learning.

Another ongoing effort is to evaluate the effects of the temporal resolution of the data on prediction tasks. As is shown in our experiments, a moderate location granularity is beneficial to the prediction tasks. Similarly, we are interested in whether a moderate temporal granularity can be beneficial to the prediction tasks.

Acknowledgments

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under the grant number 2020B010164002, the National Natural Science Foundation of China under the grant number 61725201, the Beijing Outstanding Young Scientist Program under the grant number BJJWZYJH01201910001004, and the Alibaba Group's University Joint Research Program. The work of Qiaozhu Mei was supported in part by the National Science Foundation under grant number 1633370. Huoran Li and Fuqi Lin are co-primary authors.

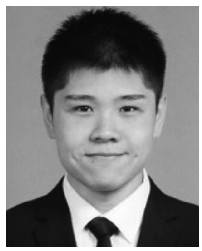
References

[1] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 5, 2014, Art. no. 2.

[2] M. Ozer, I. Keles, H. Toroslu, P. Karagoz, and H. Davulcu, "Predicting the location and time of mobile phone users by using sequential pattern mining techniques," *The Comput. J.*, vol. 59, 2016, Art. no. bxv075.

[3] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "WhereNext: A location predictor on trajectory pattern mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 637–646.

[4] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2011, pp. 34–43.
[5] J. Cao, S. Xu, X. Zhu, R. Lv, and B. Liu, "Efficient fine-grained location prediction based on user mobility pattern in LBSNs," in *Proc. 5th Int. Conf. Adv. Cloud Big Data*, 2017, pp. 238–243.
[6] M. Chen, Y. Liu, and X. Yu, "NLPMM: A next location predictor with Markov modeling," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2014, pp. 186–197.
[7] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proc. 1st Workshop Meas. Privacy Mobility*, 2012, Art. no. 3.
[8] J. Ye, Z. Zhu, and H. Cheng, "What's your next move: User activity prediction in location-based social networks," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 171–179.
[9] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed Markov-Chain model," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2011, pp. 25–33.
[10] R. Yu, X. Xia, S. Liao, and X. Wang, "A location prediction algorithm with daily routines in location-based participatory sensing systems," *Int. J. Distrib. Sensor Netw.*, vol. 2015, 2015, Art. no. 6.
[11] Z. Liu, L. Hu, C. Wu, Y. Ding, and J. Zhao, "A novel trajectory similarity-based approach for location prediction," *Int. J. Distrib. Sensor Netw.*, vol. 12, pp. 1–13, 2016.
[12] W. Li, S.-X. Xia, F. Liu, and L. Zhang, "Hybrid Markov location prediction algorithm based on dynamic social ties," *IEICE Trans. Inf. Syst.*, vol. E98.D, pp. 1456–1464, 2015.
[13] Q. Huang, "Mining online footprints to predict user's next location," *Int. J. Geographical Inf. Sci.*, vol. 31, pp. 523–541, 2017.
[14] J. Jiang, C. Pan, H. Liu, and G. Yang, "Predicting human mobility based on location data modeled by Markov chains," in *Proc. 4th Int. Conf. Ubiquitous Positioning Indoor Navigation Location Based Serv.*, 2016, pp. 145–151.
[15] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "GMove: Group-level mobility modeling using geo-tagged social media," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1305–1314.
[16] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden Markov models," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 911–918.
[17] M. Chen, X. Yu, and Y. Liu, "Mining object similarity for predicting next locations," *J. Comput. Sci. Technol.*, vol. 31, pp. 649–660, 2016.
[18] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.
[19] J. Feng *et al.*, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1459–1468.
[20] D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 2411–2414.
[21] F. Wu, K. Fu, Y. Wang, Z. Xiao, and X. Fu, "A spatial-temporal-semantic neural network algorithm for location prediction on moving objects," *Algorithms*, vol. 10, 2017, Art. no. 37.
[22] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2605–2611.
[23] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. World Wide Web Conf.*, 2019, pp. 2750–2756.
[24] Y. Jia, Y. Wang, X. Jin, and X. Cheng, "Location prediction: A temporal-spatial Bayesian model," *ACM Trans. Intell. Syst. Technol.*, vol. 7, 2016, Art. no. 31.
[25] F. Calabrese, G. Di Lorenzo , L. Liu, and C. Ratti, "Estimating origin-destination flows using mobile phone location data," *IEEE Pervasive Comput.*, vol. 10, no. 4, pp. 36–44, Apr. 2011.
[26] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2011, pp. 1082–1090.
[27] M. De Nadai, A. Cardoso, A. Lima, B. Lepri, and N. Oliver, "Strategies and limitations in app usage and human mobility," *Sci. Reports*, vol. 9, no. 1, pp. 1–9, 2019.
[28] H. Li, X. Liu, W. Ai, Q. Mei, and F. Feng, "A descriptive analysis of a large-scale collection of app management activities," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 61–62.
[29] X. Liu *et al.*, "Deriving user preferences of mobile apps from their management activities," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, pp. 1–32, 2017.
[30] X. Liu *et al.*, "Understanding diverse usage patterns from large-scale appstore-service profiles," *IEEE Trans. Softw. Eng.*, vol. 44, no. 4, pp. 384–411, Apr. 2018.
[31] Y. Mirsky, A. Shabtai, L. Rokach, B. Shapira, and Y. Elovici, "Sherlock vs moriarty: A smartphone dataset for cybersecurity research," in *Proc. ACM Workshop Artif. Intell. Secur.*, 2016, pp. 1–12.
[32] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 278–282.
[33] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.
[34] X. Liu, Y. Hui, W. Sun, and H. Liang, "Towards service composition based on mashup," in *Proc. IEEE Congress Serv.*, 2007, pp. 332–339.



Huoran Li received the PhD degree from the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include mobile computing, software engineering, and applied machine learning.

Fuqi Lin is currently working toward the PhD degree in the School of Software and Microelectronics, Peking University, Beijing, China. His research interests include mobile computing, software engineering, and data mining.

Xuan Lu received the PhD degree from Peking University, China, in 2019. She is currently a postdoc research fellow with the University of Michigan, Ann Arbor, Michigan. Her research interests include data mining and software analytics.

Chenren Xu (Member, IEEE) received the PhD degree from WINLAB, Rutgers University, New Brunswick, New Jersey, was a postdoctoral fellow in Carnegie Mellon University, Pittsburgh, Pennsylvania and visiting scholars in AT&T Shannon Labs and Microsoft Research. He is currently an assistant professor with the Department of Computer Science and a member of CECA at Peking University, China where he directs Software-hardware Orchestrated ARchitecture (SOAR) Lab since 2015. His research interests include wireless, mobility, networking, and system.

Gang Huang (Member, IEEE) is currently a full professor in Institute of Software, Peking University, China. His research interests include the area of middleware of cloud computing and mobile computing.

Jun Zhang is currently a senior data science specialist in Alibaba group, China. His research interests include data mining, network mining, and machine learning.

Qiaozhu Mei (Member, IEEE) is currently a professor with the University of Michigan School of Information, Ann Arbor, Michigan. His major research interests include data mining and information retrieval.

Xuanzhe Liu (Member, IEEE) is currently an associate professor with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include software analytics, services computing, and distributed systems.