IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 4, 2019, accepted July 22, 2019, date of publication August 12, 2019, date of current version August 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932049

An Efficient Target Tracking Approach Through Mobile Crowdsensing

DONGMING LUAN[1], YONGJIAN YANG[1], EN WANG[1], QIYANG ZENG[2], ZHAOHUI LI[3], AND LI ZHOU[4]

[1]Department of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Department of Software, Jilin University, Changchun 130012, China
[3]College of Cyber Science, Nankai University, Tianjin 300000, China
[4]School of Information, Beijing Wuzi University, Beijing 100000, China

Corresponding author: En Wang (wangen@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772230, in part by the Natural Science Foundation of China for Young Scholars under Grant 61702215, in part by the China Postdoctoral Science Foundation under Grant 2017M611322 and Grant 2018T110247, in part by the Project of Special Fund for Industrial Innovation in Jilin under Grant 2017C032-1, and in part by the Beijing Intelligent Logistics System Collaborative Innovation Center Open Topic under Grant BILSCIC-2019KF-03.

ABSTRACT As the widespread of mobile devices in recent years, mobile crowdsensing (MCS) has become a powerful mechanism to produce knowledge by collecting the individual contributed sensor data. In this paper, we aim to solve the target tracking problem through mobile crowdsensing. The traditional tracking method tends to rely on photos or videos provided by pre-deployed monitors, which may consume much power resources. Different from the traditional tracking method, the tracking approach through mobile crowdsensing (TAMC) proposed in this paper utilizes the wireless communication of mobile users to collect and contribute the valuable information about the target's whereabouts. Specifically, whenever the mobile users witness the target person, they will take photos of the target person and report the location and time of witnessing the target to the platform. Due to the fact that mobile users communicate with the platform only when they witness the target, the crowdsensing network composed of mobile users can be seen as a green network. In this way, the visited location history and corresponding time sequence of the target are available through the reports of mobile users. Once a new report is uploaded to the platform, the location history is updated. Then, according to the latest report, we apply a tree-based location prediction model named XGBoost, which is a scalable machine learning system, to predict the next place to be visited by the target. Finally, we conduct extensive experiments on a large-scale real-world dataset, namely, the Gowalla check-in dataset. The experimental results show that compared with the baseline methods, the tracking approach can predict the next places accurately.

INDEX TERMS Mobile crowdsensing, target tracking, mobility prediction.

I. INTRODUCTION

Due to the fact that smartphones and iPads have been widely used in people's daily life, Mobile Crowdsensing (MCS) has emerged as a novel paradigm to carry out the task of collecting sensing data through the mobile devices. Now the smart devices are equipped with powerful sensors, which could collect various kinds of data such as picture, video, text and so on. Therefore, MCS has attracted a lot of attention in recent years and the technology advancements of many fields [1], [2]. MCS has been used to solve multiple problems ranging from roadside parking management [3], road condition detection [4], environmental pollution monitoring [5] and urban WiFi characterization [6] to digital map updating [7]. In general, a MCS system consists of three components: task requester, service platform and mobile user. Firstly the task requestor releases the task to the platform and then the platform is responsible to allocate the task to the mobile users who register on the service platform. Next, the mobile users move to the task location to perform the task and upload the data to the platform. Finally, the service platform analyzes and processes uploaded data. With the help of such a platform, in this paper, we aim to solve the target tracking problem under the mobile

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

crowdsensing scenario. Current target tracking systems are mainly based on pre-deployed video monitors, but in most of the rural and suburb areas, there are few monitors. Even in city, the monitor network may not be deployed densely and its coverage is also limited. Moreover, the image recognition performance of the monitor is affected by many factors such as weather and image resolution. In view of this phenomenon, we try to solve the target tracking problem with the help of MCS. Specifically, when some people or organizations want to track the location of someone (target person), they will publish the task to the service platform as the task requesters. Then, the platform will allocate the task to the mobile users.

In this case, the platform could utilize the mobile users who are in the same city with the target person to collect information. When the mobile users witness the target person, they will take photos of the target person and upload the information to the platform including the time and the location they meet. Subsequently, the service platform can determine whether the users have identified the right target person by the uploaded photos and construct a location history list of the target person consisting of a series of place names and corresponding time. When a new report is uploaded to the service platform, the location history list will be updated.

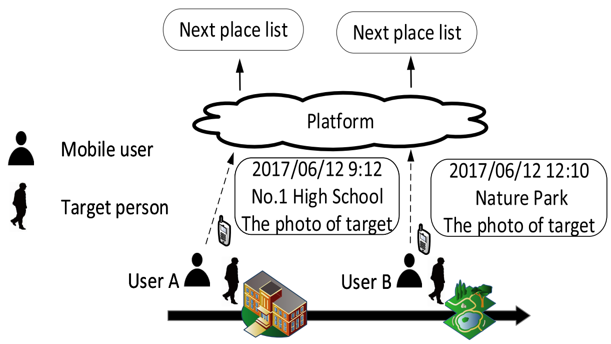


FIGURE 1. **The scenario of tracking target movement through mobile crowdsensing.**

Intuitively, the target is very likely to be near the latest reported area. However, due to the mobility of the target and the delay in uploading the reports, the target probably has left the location of the last report in the present moment. To cope with this, we utilize a location prediction model to predict the next location that the target will visit. As shown in Fig. 1, when mobile user A witnesses the target person at No.1 High School, he will take photos and report the current time '2017/06/12 9:12' and the place 'No.1 High School' where they meet to the platform, so does mobile user B. Whenever the platform receives a new report, firstly, it will determine whether the users have identified the right target person by uploaded photos. If the identification is right, the platform adds the place and time to the location history list. Then, the location prediction model will produce and update the next place list that the target person is most likely to go according to the latest report. The requester could search the target around the predicted locations.

So the essence of tracking target is location prediction. Song *et al*. [8] indicate that there exists 93% potential predictability in mobile user mobility. They prove the periodicity of human mobility. It is worth noting that a witness's report only contains one location information. The place sequence of the location history list is not consecutive, since the location in the latest report may not be the next place of the last reported place. Specifically, the target person may have visited multiple locations between the two consecutive uploaded reports. Different from conducting prediction according to a spatial-temporal location sequence, we predict the next location given the location information of the latest uploaded report. To solve this problem, we comprehensively utilize individual and global spatial-temporal data, friendship information and place attribute to predict the target movement. The place attribute here refers to the type of place, such as cinema, stadium and school. Intuitively, the individual mobility may have correlation with his friends. That is to say, a person may move to the location where his friends have ever been. Another challenge emerges that how to utilize the information from multiple aspects mentioned above comprehensively. we extract various kinds of features collaboratively and then put them into a supervised classification model, named XGBoost [9] to predict target movement. Explicitly, we regard all the places in the city where the target person appears often as the candidate place set and use the trained model to calculate the probabilities that the target person will arrive at the candidate places. Finally, the approach ranks the candidate places in a descending order of the probabilities and returns a top-k next place list.

In conclusion, the main contributions in this paper are summarized as follows:

- We put forward the target tracking problem under mobile crowdsensing scenario. Unlike most traditional tracking approaches which are based on pre-deployed video cameras, we aim to solve the target tracking problem by means of mobile users, who will take photos and report the location and time information to the system when they witness the target.
- We extract features from multiple aspects: spatial-temporal information, friendship information and place attribute. Then, we utilize them to train the supervised classification model collaboratively.
- To efficiently predict the target's location, we address the location prediction problem as a binary classification problem and utilize a supervised learning model named XGBoost, which is a scalable machine learning system, to judge whether a place in the candidate location list will be visited by the target person. Furthermore, the model will give the probability that the target person will arrive at each candidate location. Through sorting the probabilities, we could get the top-k locations that are most likely to be visited by the target.
- We conduct extensive experiments on a real-world dataset to evaluate the performance of the proposed approach. The experimental results show that the

proposed approach is superior to the baseline methods on various metrics.

The remainder of our paper is organized as follows. In Section II, we review the related works. Section III describes and formulates the problem. In Section IV, we give the description about feature extraction. Section V introduces the theoretical principle of the location prediction model in detail. In Section VI, we demonstrate the experiment results. Finally, in Section VII, we conclude the paper.

II. RELATED WORKS

A. TRADITIONAL TARGET TRACKING

The traditional method of tracking target is by the means of pre-deployed video monitors. There are several literatures about tracking target in the wireless sensor network. Liu *et al*. [10] put forward a dynamic node cooperation prediction framework in wireless video sensor network. In their paper, they build a nonlinear localization-oriented sensing model and apply the Sequential Monte Carlo algorithm (SMC) to compute the target location. The proposed node cooperation framework can balance the tradeoff between the performance of the framework and the total cost of the algorithm. In [11], the authors aim to solve the minimum camera barrier coverage problem in wireless sensor networks. They model the full-view-covered regions and their relationships as a directed graph and propose an algorithm based on the graph. Furthermore, they prove the correctness of the solution.

B. TARGET TRACKING IN MOBILE CROWDSENSING

There are some works about tracking target under mobile crowdsensing scenario. Shin *et al.* [12] develop a mobile application to expedite the process of finding missing children by means of crowd cooperation. However, the application needs to coordinate with the special nametag carried with children. This way increases the cost of system and affects the availability negatively. In [13], the authors develop a crowdsensing-based object tracking system (CrowdTracker) for the purpose of tracking vehicles on road. Visual Crowd Sensing (VCS) is applied to the tracking system which utilizes participants' contributed photos with a lower incentive cost. Cao *et al.* [14] put forward a target tracking framework in wireless sensor network based on crowdsourcing, explicitly, they design a market mechanism for sensors to exchange information in order to track object on assumption that the sensors are selfish and profit-incentive. In general, the literatures mentioned above pay no attention to tracking human through predicting human mobility. Different from these studies, in this paper, we track the target person by predicting human mobility.

C. LOCATION PREDICTION

It is of great importance to predict the target movement in order to track target trajectory. There are a lot of works that address the future location prediction problem. As the Markov model can address the time sequence data, it is widely used to build the location prediction model. Kim *et al.* [15] propose a location prediction framework to predict user destination based on hidden Markov model. They use k-nearest neighbor and decision tree to recognize user's current location and have developed a prediction system on smart phones. In [16], the authors analyze the characteristic of human mobility and build a hybrid Markov model considering the time distribution of trajectory patterns. In order to improve the performance of the prediction model, they take the time when location is visited into consideration. In [17], the authors design a system utilizing a hidden Markov model in order to predict the users' destinations when they start a new trip based on the GPS log history. Moreover, the approach proposed produces a street-map for each user to improve the performance of the system. In recent years, neural network has received more and more attention and some researchers have applied it to solve location prediction problem. Parija *et al.* [18] put forward a neural network model using back propagation technique to predict the position of mobile user in cellar network environment. Kim *et al.* [19] put forward a vehicle route prediction model which utilizes recurrent neural network called long short-term memory to analyze the temporal behaviors of the vehicle from a large amount of data.

The Markov model is appropriate to process the relatively dense spatial-temporal data with consecutive place trajectory. However, when facing data with sparse location points, the performance of Markov chain based model is not satisfactory in most cases. Although neural network model is capable of dealing with the sparse data in the training process, the process of training the neural network model needs a huge number of data and the design of the neural network is complex. The tracking approach proposed in this paper can predict the future location given only one mobility record, moreover, the data size required in training process is not as large as the neural network.

III. PROBLEM DESCRIPTION AND FORMULATION

As mentioned before, whenever the mobile user finds the appearance of the target person, he will take photos and report the location and time of witnessing the target person. Then, the location history list is updated and the requester can search the target with the help of the reported list. Compared with the way that only provides the latest reported location, this approach can provide more valuable information to analyze and judge the possible next places of the target. Obviously, the location prediction model plays an important role during the predicting process.

We assume that the target person's historical mobility record and friendship information are available for the platform. It is worth noting that the target's historical mobility records are different from the location history list. The location history list consists of place and time series reported by mobile users. The historical mobility records keep the individual's mobility information in the past period of time. According to the latest record in the location history list, we extract individual mobility features from the historical mobility record. The friends mentioned in the paper are the target's friends on social network. The friendship information contains the friends' historical mobility records. Another assumption is that the total historical mobility records are available on the platform by means of large-scale crowdsourcing. So we can extract global mobility features to help predict the future location of the target. The platform can obtain the corresponding type of place easily, such as school and restaurant, so we can extract the features from it. The assumptions mentioned above are practical when the government or the police want to track somebody when traditional tracking methods perform unsatisfactorily. The police can access the target's historical mobility records and friendship information, so they can track the target with the help of the tracking approach proposed in this paper.

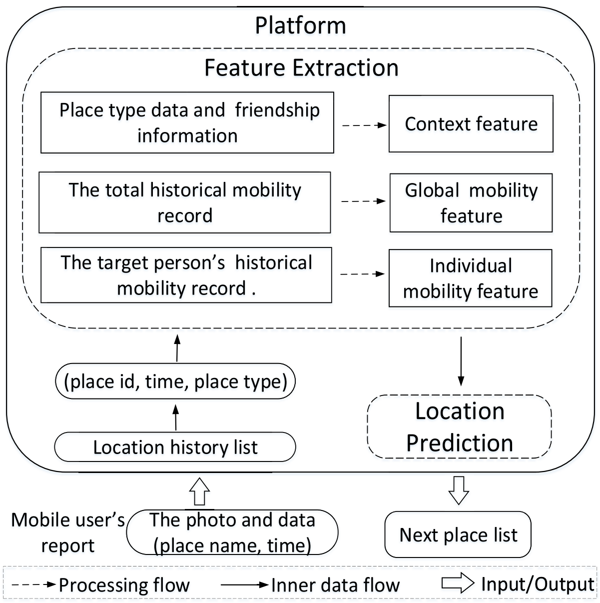


FIGURE 2. The illustration of prediction platform.

As shown in Fig. 2, when a mobile user finds the target person, a new report is uploaded to the service platform. It is added to the location history list firstly. Then, the platform replaces the place name in the report with a unique place ID and adds the item of place type in report. Subsequently, the features from the target's historical mobility record, the total historical mobility record and place type as well as friendship information are extracted respectively. Finally, the features are put into the location prediction model to predict the possible next places. It is worth noting that the corresponding candidate place set is determined through selecting all the places of the city where the target person resides. Specifically, we can select all the places in the large-scale crowdsourcing data in the given city. For each candidate place, the model will predict the probability to be visited by the target. The platform returns a next place list including top-k places most likely to be visited in the end.

For each target person u, assume that the report about u's location after processed by platform is a triple $\langle p, t, s \rangle$. Specifically, its latest reported place ID and time are defined as p, t and the place type is s. We define the candidate place set of u as L_u. Then we formulate the next place prediction problem as follows. Given a triple $\langle p, t, s \rangle$ of the latest report reported by the mobile user, we try to predict the next places probably to be visited by the target person u. For each place $l \in L_u$, we need to judge whether l will be visited by the target person and compute the probability that the target person will visit l in the future. Finally we rank the set of L_u according to the probabilities and select the top-k places most likely to be visited.

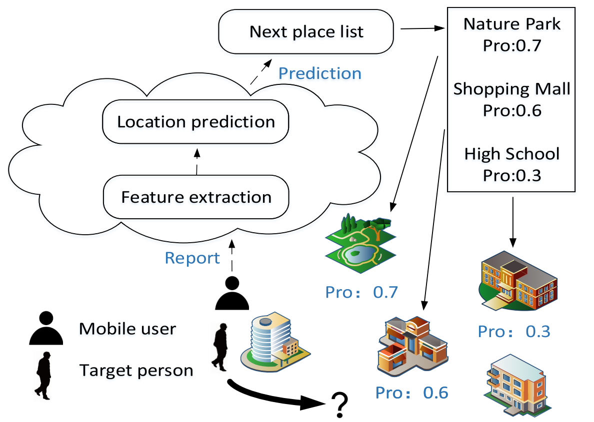


FIGURE 3. The illustration of target tracking. A mobile user finds the target appearing near a building. Then the user sends the report to the platform. The platform will extract features and predict the next places. In the next place list, the probabilities of visiting school, park and shopping mall are 0.3, 0.7 and 0.6 respectively.

To solve this prediction problem, we formulate it as a binary classification problem and apply a binary classification strategy in the paper. Then for one candidate place, we will judge whether the place will be visited by the target. As shown in Fig. 3, when the mobile user witnesses the target, a new report about the target person will be uploaded to the platform, the triple $\langle p, t, s \rangle$ is updated. Then, the features are extracted and the algorithm is used to predict the next location where the target will be according to the new triple. For each place in target's candidate place set, the prediction model will provide the probability that it will be visited by the target. In Fig. 3, the platform produces a next place list which contains the places probably to be visited by the target. In the list, the probabilities of visiting school, park and shopping mall are 0.3, 0.7 and 0.6 respectively. Since it is a binary classification task, so it is worth noting that the probabilities of each candidate place are independent, that is to say, the sum of them is not equal to 1. The main notations used through the paper are illustrated in Table 1.

IV. FEATURE EXTRACTION

As mentioned above, we will utilize multiple aspects of data to collaboratively predict the next place. To this end, we use

TABLE 1. Main notations used throughout the paper.

Symbol	Meaning
$\langle p, t, s \rangle$	The place ID, time and place type in the report after processed by the service platform
L_u	The candidate place set of individual u
R_u	The historical mobility record of individual u
R	The historical mobility record of all individuals
$\hat{y}_i, y_i$	The predicted result of i-th example, the real result of i-th example
M	The number of regression trees in the system
E	The number of a tree's leaves
d	The score of corresponding leaves in the tree
$\mathcal{L}^{(k)}$	The objective function in k-th iteration
$l(\hat{y}_i, y_i)$	The loss function of the prediction model measuring the difference between $\hat{y}_i$ and y_i
$\hat{y}_i^{(k)}$	The prediction result of the i-th sample at the k-th iteration.
f_m	The output of m-th regression tree
$\Omega(f)$	The regularization function of f

a prediction model named XGBoost, which is a scalable machine learning system. The most important factor of its success is its scalability in nearly all scenarios. We need to extract features from the individual historical mobility record, the total historical mobility record and place type as well as friendship information to train the prediction model and produce the prediction results.

In this section, we give a detailed description of features used to solve the next place prediction problem. We divide them into three categories: individual mobility feature, global mobility feature and context feature. Individual mobility feature is extracted from the target's mobility information, while global mobility feature is extracted from large-scale crowdsourcing data of mobile users, it reflects the common rules about human mobility. Besides, context feature is relevant to friendship data and the place type. We denote l_0 and t_0 as the target's location and time reported by the mobile user. Moreover, the historical mobility record of individual u is expressed as $R_u = \{r_1, r_2, \ldots, r_n\}$, where $r_i = \langle p, t, s \rangle$. Each item in the list R_u is in chronological order. Total historical mobility records of all individuals are defined as R. Let U be the set of all individuals. For individual u, let (r_f, r_n) denote the two consecutive records in R_u.

A. INDIVIDUAL MOBILITY FEATURE

Many researches [20], [21] have shown that human mobility has a strong periodicity, which means that people tend to appear in the same location around the similar time of the day. The features can provide insightful and significant information to predict the possible next place.

1) INDIVIDUAL PLACE TRANSITION

Here we consider the time periodicity of human movement. Intuitively, it is very important to consider individual direct transition among places. This feature is represented by the number of the direct transitions from l_0 to the place l_c of the target in the past [22]. This value can be obtained by counting the past direct transitions in the target person's historical mobility data, which is defined as Eq. (1).

$$Place\_trans = |\{(r_f, r_n) : r_f, r_n \in R_u : r_f.p{=}l_0{\wedge}r_n.p{=}l_c\}| \tag{1}$$

2) WEEK MODE VISIT

There is a significant difference in human behavior among different days of a week. For week mode, $w(t) = \{1, 2, 3, 4, 5, 6, 7\}$, which represents the day of the week from Monday to Sunday, respectively. $w(t_0)$ represents the day of the week of the reported time t_0 of witnessing the target, then we extract week mode visit feature, which measures the number of visits in place l_c in $w(t_0)$ of target u, as shown in Eq. (2).

$$Week\_visit = |\{(p, t, s) \in R_u : w(t) = w(t_0) \wedge p = l_c\}| \tag{2}$$

3) DAY MODE VISIT

To model human mobility in the different time of a day, we divide the period of a day into four parts: before dawn (0:00-6:00), morning (6:00-12:00), afternoon (12:00-18:00), night (18:00-24:00). For day mode, we map the time into discrete time windows. $d(t){=}\{1, 2, 3, 4\}$ represents the above divided four parts respectively. Specifically, $d(t_0)$ represents the corresponding part in the day mode of the reported time t_0 of witnessing the target. The feature models the number of visits in place l_c in $d(t_0)$ of target u, which is defined as Eq. (3).

$$Day\_visit = |\{(p, t, s) \in R_u : d(t) = d(t_0) \wedge p = l_c\}| \tag{3}$$

4) TRANSITION TIME

This feature represents the average elapse time spent by the target u from current place l_0 to the place l_c. For place l_c, let $R_c = \{(r_f, r_n) : r_f, r_n \in R_u, r_f.p = l_0, r_n.p = l_c\}$ We have:

$$Trans\_time{=}\frac{\sum_{(r_f, r_n)\in R_c} r_n.t - r_f.t}{|R_c|} \tag{4}$$

B. GLOBAL MOBILITY FEATURE

In most cases, people tend to follow common paths [23]. For example, children always go to school on similar routes and the public bus also follows the same route. This property has been used by many literatures to predict the next place. Now we demonstrate how we extract the global mobility feature from the global information to help solving the next place prediction problem. The details are shown as follows.

1) NUMBER OF VISITORS

This feature measures the popularity from the global view. There are some places, such as train station, tourist spot, which will always attract many people to go there. Based on the assumption that a person may follow the crowd, the feature counts the number of distinct visitors that have visited place l_c before, which is formally expressed as Eq. (5).

$$Visitor\_num = |\{u \in U : r \in R_u, l_c = r.p\}| \tag{5}$$

2) NUMBER OF VISITS

This feature cooperatively works with the feature *Number of visitors*. This feature counts the number of total visits to the place l_c of all visitors and is also a measurement of popularity, which is defined as Eq. (6).

$$Visit\_num = |\{r \in R : r.p = l_c\}| \tag{6}$$

C. CONTEXT FEATURE

To improve the performance of the model, we also extract some other context features from friendship information and the place type. The previous works have found that the significant influence of the friendship to human mobility. For instance, in [24], the authors find that mobile users' behaviors are affected by their social friendship from spatial and temporal dimensions in location based social network. In [25], the authors find that it is a great possibility for an individual to move to where a friend has visited previously. The authors in [26] use the place type to infer the people's current activities and further solve the next place prediction problem. The exhaustive description is shown as follows.

1) NUMBER OF FRIENDS' VISITS

As mentioned above, an individual may go to the place where the friends have visited. This feature calculates the number of the target person's friends' visits in place l_c, which is shown in Eq. (7). Here, we denote $F(u)$ as the friend set of u.

$$Friend\_visit = \sum_{u' \in F(u)} |\{(p, t, s) \in R_{u'} : p = l_c\}| \tag{7}$$

2) PLACE TYPE TRANSITION

To the case of the direct movement, measuring the place type transition of the target person is meaningful to help improving the performance of the prediction model. This feature is defined as follows.

$$Type\_trans = |\{(r_f, r_n) : r_f, r_n \in R_u \wedge r_f.s = l_0.s \wedge r_n.s = l_c.s\}| \tag{8}$$

Given a reported location for prediction, we have a candidate place set. The features calculate a numeric value for the candidate place and all the features of a place form a feature vector. The feature vector is the input of the location prediction model. That is to say, the candidate place information is input to the prediction model in the form of the feature vector. Then, the model will calculate the probability of the candidate place to be visited by the target.

V. SUPERVISED CLASSIFICATION MODEL

In this section, we put the previous features into a supervised learning model. As mentioned before, we formulate the next place prediction problem into a binary classification problem. Then for each place in the candidate place set, the model calculates the probability that the target person will visit in the future and ranks the probabilities in descending order, so that we could select the top-k places most likely to be visited.

Just as other machine learning systems, the prediction model needs the training process to adjust the parameters and model structure. The data used to train the model is called training set. To construct the training set, for the two consecutive records in historical mobility record, we regard the place in the former record as the current location and the place in the latter record as the next place. We build the positive samples by extracting the features according to the two consecutive records and add a positive label to it. Then, to build the negative samples, we randomly select ten other places that the target person has not visited before, extract features and add negative labels to them. In this way, the extracted features are mapped to the corresponding samples. The training process aims to teach the model what the important characteristics are and how to adjust the structure and parameters to get a good prediction performance. As there also exist some parameters in the model that need to set manually, we use the validation set to evaluate the model and parameters. The validation set is built in the same way as the training set. it is worth noting that the training set and the validation set are disjoint.

We utilize a tree-based prediction model, named XGBoost, to learn the way in which the feature vectors correspond to the positive and negative labels. XGBoost is a powerful and scalable machine learning system, which is a tree boosting system based on multiple regression trees.

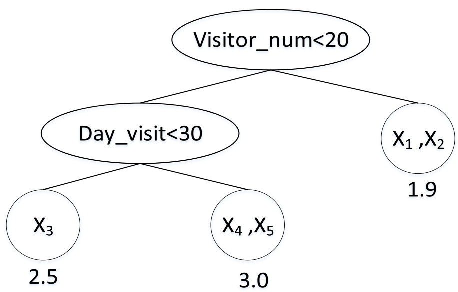


FIGURE 4. An example of regression tree.

Now, we will give a detailed description about how XGBoost works. First, we give a brief introduction about the structure of regression tree. The node in the regression tree except the leaf node represents a feature which splits the sample into two parts. Unlike decision tree, the leaf of tree contains a continuous score. As Fig. 4 displays, the tree contains two features: *Visitor_num* and *Day_visit*. Their corresponding split points are 20 and 30. The three leaf nodes contain sample set (x_1, x_2), (x_3), (x_4, x_5) and correspond to score 1.9, 2.5, 3.0, respectively. By means of regression tree, the given example is mapped to a leaf index and a corresponding score.

XGBoost is a tree boosting system which is based on multiple regression trees. As shown in Fig. 5, the final prediction

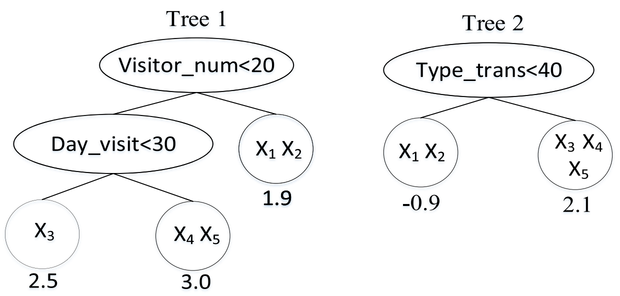


FIGURE 5. An example of tree boosting system. The final result is the sum of all the trees' prediction results, that is to say, $\hat{y}_1 = 1.9 - 0.9 = 1$.

results are obtained by the sum of the output score of each regression tree. As shown in Eq. (9), for a given dataset $D=\{(x_i, y_i)\}$ with n examples, where x_i is the input of the model obtained by encoding the value of features and y_i is the label of the example. The model predicts the output result by applying additive function:

$$\hat{y}_i = \sum_{m=i}^{M} f_m(x_i), \tag{9}$$

where $\hat{y}_i$ is the prediction result, M is the number of regression trees in the system and f_m corresponds to the output of m-th regression tree. As mentioned above, the final prediction result is the sum of regression trees' results, which are calculated by the scores of the corresponding leaves in each tree (denoted by d). In order to learn the function of the model, the objective is shown as follow.

$$\begin{aligned} minimize\ \mathcal{L} &= \sum_i l(\hat{y}_i, y_i) + \sum_m \Omega(f_m) \\ \Omega(f) &= \alpha E + \frac{1}{2}\lambda ||d||^2 \end{aligned} \tag{10}$$

Here $l(\hat{y}_i, y_i)$ measures the difference between the prediction result $\hat{y}_i$ and the true label y_i which is a differential convex function. E is the number of a tree's leaves. α and λ are two constants. Ω is used for regularization. Actually, it is used to avoid over-fitting by training a model applying simple functions. The more complex the model is, the larger Ω will be. In conclusion, Ω can be regarded as a penalty of the complex model.

Let $\hat{y}_i^{(k)}$ signify the prediction result of the i-th sample at the k-th iteration. Due to the fact that the prediction result is obtained by additive approach, so $\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + f_k(x_i)$ and $\hat{y}_i^{(0)} = 0$. This means that, according to Eq. (10), we greedily add f_k that can most enhance the performance in each iteration. Let $\mathcal{L}^{(k)}$ represent the objective function in k-th iteration which is calculated as Eq. (11). The model needs to add f_k to minimize it.

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(k-1)} + f_k(x_i)) + \Omega(f_k) \tag{11}$$

Then, according to the second-order Taylor expansion Eq. (12) is obtained.

$$\mathcal{L}^{(k)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(k-1)}) + c_i f_k(x_i) + \frac{1}{2} z_i f_k^2(x_i)] + \Omega(f_k) \tag{12}$$

where $c_i = \frac{\partial l(y_i, \hat{y}_i^{(k-1)})}{\partial \hat{y}_i^{(k-1)}}$ and $z_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(k-1)})}{\partial \hat{y}_i^{(k-1)}}$, by removing the constant term, a simplified version of objective at k-th iteration is represented as follow.

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n} [c_i f_k(x_i) + \frac{1}{2} z_i f_k^2(x_i)] + \Omega(f_k) \tag{13}$$

Let V_j denote the sample set that map to leaf j and according to Eq. (10), Eq. (13) can be rewritten as follow.

$$\begin{aligned} \mathcal{L}^{(k)} &= \sum_{i=1}^{n} [c_i f_k(x_i) + \frac{1}{2} z_i f_k^2(x_i)] + \alpha E + \frac{1}{2}\lambda \sum_{j=1}^{E} d_j^2 \\ &= \sum_{j=1}^{E} [(\sum_{i \in V_j} c_i) d_j + \frac{1}{2}(\sum_{i \in V_j} z_i + \lambda) d_j^2] + \alpha E \end{aligned} \tag{14}$$

According to the structure of a regression tree, the optimal score d_j^* of a leaf j can be calculated by Eq. (15). Let $Q = (\sum_{i \in V_j} c_i) d_j + \frac{1}{2}(\sum_{i \in V_j} z_i + \lambda) d_j^2$. And Eq. (15) is obtained by $\frac{\partial Q}{\partial d_j} = 0$.

$$d_j^* = -\frac{\sum_{i \in V_j} c_i}{\sum_{i \in V_j} z_i + \lambda} \tag{15}$$

So the corresponding optimal value of $\mathcal{L}^{(k)}$ is shown as follow.

$$\mathcal{L}^{(k)} = -\frac{1}{2} \sum_{j=1}^{E} \frac{(\sum_{i \in V_j} c_i)^2}{\sum_{i \in V_j} z_i + \lambda} + \alpha E \tag{16}$$

Eq. (16) can be regarded as a score function to evaluate the performance of the regression tree. Due to the fact that there is no way to enumerate all the tree structures, hence, the model adopts a greedy algorithm. Specifically, it starts from a single node then splits the node and adds the branches to the tree iteratively. Let V_L and V_R signify the sample sets of left and right nodes, respectively. Let $V = V_L \cup V_R$, then the decrease of the loss after the split is shown as follow.

$$G = \frac{1}{2}[\frac{(\sum_{i \in V_L} c_i)^2}{\sum_{i \in V_L} z_i + \lambda} + \frac{(\sum_{i \in V_R} c_i)^2}{\sum_{i \in V_R} z_i + \lambda} - \frac{(\sum_{i \in V} c_i)^2}{\sum_{i \in V} z_i + \lambda}] - \alpha \tag{17}$$

In each iteration that the node is divided, the model chooses the feature and split point that make G largest, then continues to split the child node through the above approach until the max depth of the tree reaches the given threshold. Finally, all the trees are built one by one. It is worth noting that the probability that a candidate place will be visited by the target person is calculated by mapping the final score into a sigmoid function.

To find the best feature and the split point in each iteration, one basic feasible approach is to enumerate over all possible features and split points greedily. In order to accelerate the greedy process, the algorithm sorts the data according to the feature values and enumerates data in the sorted order. However, when the size of data is large, it needs high computational and memory demanding. In the worst situation, the data may be too large to be put into the memory. The greedy algorithm is not practical in this case. A weighted quantile sketch algorithm [9] is used in this situation, which is a kind of approximate algorithm. Its basic idea is extracting a portion of split points according to the quantiles of features' distribution without enumerating all split points.

In conclusion, XGBoost is a tree-based boosting system with the help of multiple regression trees. This is an efficient and widely used machine learning approach.

VI. EXPERIMENT EVALUATION

A. DATA PREPARATION

In this paper, we use a real-world dataset to carry out the extensive experiment for evaluating the performance of the proposed approach. The dataset [27] was collected from a location based social network site, called Gowalla, which contains more than 600,000 users. The dataset contains more than thirty million check-in records made by 319,063 users over 2,844,076 locations. According to the type of place, the places in the dataset have been classified into seven root categories, such as community, food, outdoors, shopping and so on. Each of them contains some subcategories. An example of the distribution of check-in points is shown in Fig. 6.

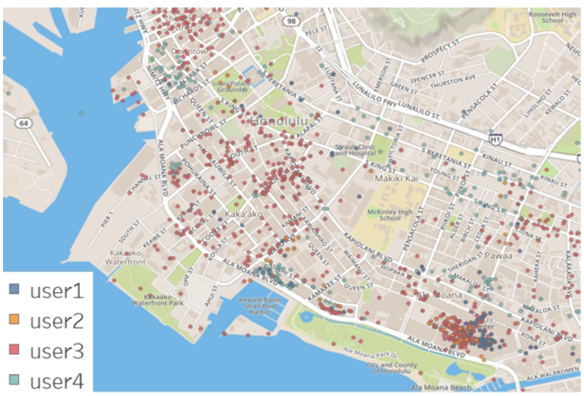


FIGURE 6. An example of check-in points of four users in Gowalla dataset.

To prove that the result of the proposed model does not depend on the specific dataset, we choose check-in records in two cities, i.e., Honolulu and London as experiment datasets. The detailed parameters of selected datasets are shown in Table 2. The time scope of the check-in data is three years ranging from 2009 to 2011 and the number of check-in records in 2010 is much larger than that of 2009 and 2011. Hence, we select check-in records in 2010 and delete the places with no type information in experiment.

TABLE 2. Dataset description.

Dataset	Honolulu	London
Users	198	523
Check-in records	203,982	216,157
Places	8,561	20,765

TABLE 3. The confusion matrix.

Real situation \ Prediction result	Positive	Negative
Positive	TP	FN
Negative	FP	TN

B. EVALUATION METRICS

In order to evaluate the performance of the proposed model, we measure the following metrics in this paper. Firstly, we introduce some definitions here, we denote *TP* (True Positive) as the number of positive samples that are classified into the positive category and *FP* (False Positive) as the number of negative samples that are classified into the positive category. While *TN* (True Negative) signifies the number of negative samples that are classified into the negative category and *FN* (False Negative) signifies the number of positive samples that are in negative category. The confusion matrix of the classification result is listed.

Precision: $P = \frac{TP}{TP+FP}$. It represents the percentage of correctly predicted samples in the positive prediction.

Recall: $R = \frac{TP}{TP+FN}$. It is the fraction of the correctly predicted positive samples in the total number of positive samples.

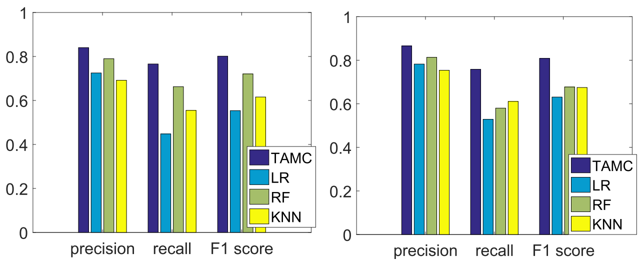
$F1$ **score:** $F1 = \frac{2RP}{R+P}$. $F1$ score is defined based on the harmonic mean of precision and recall metrics: $\frac{1}{F1} = \frac{1}{2}(\frac{1}{P}+\frac{1}{R})$. It is a comprehensive evaluation metric of precision and recall metrics.

Accuracy@N: we assume that the prediction is successful if the next place is in the predicted list, when the predicted list size is N. Accuracy@N represents the percentage of successful prediction instances in the total predictions.

Average Percentile Rank (APR): the Percentile Rank [28], i.e., PR is computed as follows: $PR = \frac{|L|-rank(k)+1}{|L|}$, where $|L|$ is the candidate list size and $rank(k)$ is the ranking of the place k in the prediction list after sorting all the places in descending order of the probabilities to be visited. The PR is equal to 1 when the place is ranked first. The Average Percentile Rank (APR) is obtained by calculating the average value of all correct predictions and it measures the normalized position of the correct predictions.

C. RESULTS AND EVALUATION

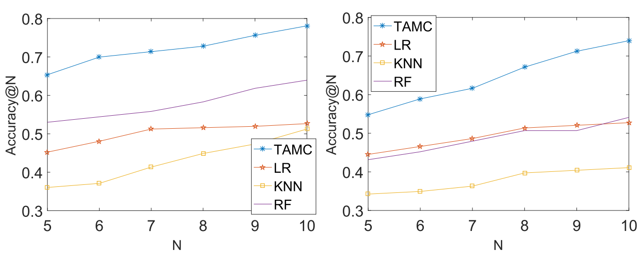
The historical mobility records are divided into four parts, the recording period is set as one year. Explicitly, we use the data from July to September as the training set and the data from October to November as validation set. Finally, we use the data in December as testing set and the data from January to June are used to extract features.



(a) Honolulu (b) London

FIGURE 7. Performance comparison about precision, recall and $F1$ score.

The baseline methods considered in our experiments are Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbor (KNN). They are all widely used and classical classification algorithms. First, we compare the prediction performance of TAMC proposed in this paper with the baseline methods in the validation set. The results are shown in Fig. 7. It is obvious that the proposed approach outperforms the baseline methods in terms of precision, recall and $F1$ score in both Honolulu and London datasets. Explicitly, $F1$ score of TAMC is 8% and 13% higher than that of Random Forest (the best baseline method) in Honolulu and London datasets, respectively. The precision metric of TAMC is 5% higher than that of the best baseline method in both datasets. In Honolulu dataset, the precision metric of TAMC is 15% higher than that of the worst baseline method. The recall metric of TAMC is 10% and 14% higher than that of the best baseline method in Honolulu and London datasets respectively. In Honolulu dataset, the recall metric of TAMC is 32% higher than that of the worst baseline method.



(a) Honolulu (b) London

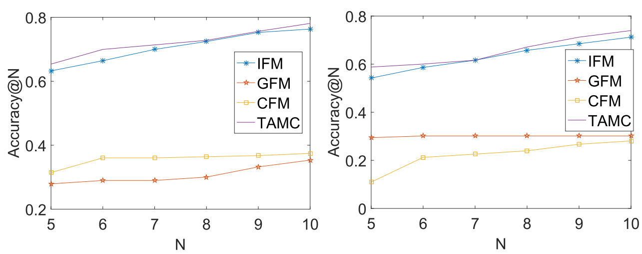
FIGURE 8. Accuracy@N comparison results on different models.

Next, we evaluate the prediction performance by comparing Accuracy@N and Average Percentile Rank (APR) which demonstrate the effectiveness of the prediction model. Performance comparisons are displayed in Fig. 8 and Table. 4. As is shown in Fig. 8, Accuracy@N of all the methods grow with the increase of the prediction list size N. The performance of TAMC is superior to the baseline methods. Specifically, the value of TAMC is at least 0.1 higher than that of the best baseline method in both Honolulu and London dataset. KNN is the worst method in both two datasets. When the prediction list size N is 6, the value of TAMC is 0.32 higher than that of KNN in Honolulu dataset. It is evident that when the prediction list size is 10, the disparity of the value of TAMC between the best baseline method is the most which is up to 0.19 in London dataset. When the prediction list size is 9, the Accuracy@N value of TAMC exceeds 0.7 on both Honolulu and London datasets, which are 0.75 and 0.71, respectively. Table. 4 plots Average Percentile Rank about different methods on two datasets. We can see that in London dataset, the APR values of LR and RF are close to that of TAMC, while the disparity becomes large in Honolulu dataset. Specifically, TAMC is 0.99 in two datasets, the result is 0.3 higher than RF and 0.6 higher than LR. As there exist thousands of places in the candidate place set in each prediction, it means that the average rank of the correctly predicted places in TAMC is higher than that of other baseline methods at least dozens of positions.

TABLE 4. Average percentile rank on different models.

Dataset \ Model	TAMC	LR	RF	KNN
Honolulu	0.993	0.937	0.965	0.915
London	0.99	0.98	0.98	0.94

In order to prove the effectiveness of applying the combined features, we evaluate the performance of TAMC with combined features in contrast to the three baseline methods. Individual Feature based Method (IFM) means the baseline method that only uses individual mobility features and Global Feature based Method (GFM) represents the baseline method that only takes global mobility features into consideration, while Context Feature based Method (CFM) signifies the baseline method using the context features mentioned before. Accuracy@N and APR comparison results between different baseline methods are displayed in Fig. 9 and Table. 5.



(a) Honolulu (b) London

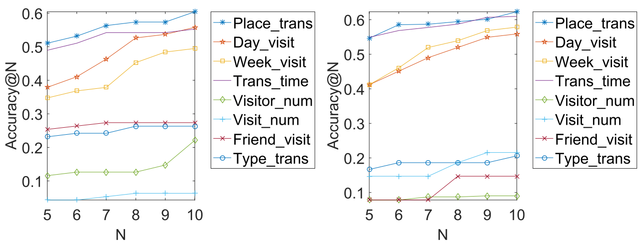
FIGURE 9. Accuracy@N comparison results on different kinds of features.

Fig. 9 plots Accuracy@N comparison on different kinds of features along with the increase of the prediction list size N. In Fig. 9-(a), when N is equal to 8 and 9, though the value of TAMC and IFM coincide approximately, the value of TAMC is still higher than IFM. The value of IFM is the highest in London dataset when the next predicted location list is small, but as the list size N increases, the TAMC performs better than other baseline methods. Though the value of CFM is

TABLE 5. Average percentile rank on different features.

Dataset \ Method	TAMC	IFM	GFM	CFM
Honolulu	0.99	0.99	0.92	0.85
London	0.99	0.99	0.94	0.79

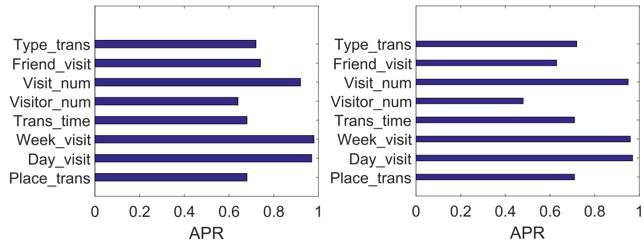
the lowest among all methods in Fig. 9-(b), as the prediction size N increases, the value of CFM increases. When the prediction size is 10, Accuracy@N of CFM approximates to the value of GFM. Table. 5 demonstrates Average Percentile Rank (APR) comparison results on different kinds of features. The APR of CFM is the worst in both Honolulu and London dataset, specifically, the value is 0.81 and 0.74 respectively. The APRs of TAMC and IFM are equal to 0.99 and are higher than the other two baseline methods. In Honolulu dataset, the difference between the highest value and the lowest value is 0.14 and in London dataset, this difference is 0.2. In general, from Fig. 9 and Table. 5, we can conclude that the individual mobility feature plays an important role in the model, while the global mobility feature and the context feature have auxiliary effects to predict the next place.



(a) Honolulu (b) London

FIGURE 10. Accuracy@N comparison results for the single feature.

Fig. 10 shows Accuracy@N performed by using single feature to predict the next place with the change of the prediction list size. From Fig. 10, we can conclude that in most cases, using the single feature does not have a good prediction performance. In Fig. 10-(a), only Accuracy@N of the feature Individual place transition (*Place_trans*) exceeds 0.6 when the prediction list size is 10. The value of feature Number of visits (*Visit_num*) below 0.1 which is the lowest among all features. The value of most features grow along with the increase of the prediction list size N except *Friend_visit*. And in Fig. 10-(b), the highest value of Accuracy@N is about 0.6, in the worst case, the value is below 0.1. The values of feature Individual place transition (*Place_trans*), Transition time (*Trans_time*) and Week mode visit (*Week_visit*) are the highest along with the increase of the prediction size N. Fig. 11 demonstrates the value of Average Percentile Rank comparison results for using the feature individually. The main observation is that the values of feature Day mode visit (*Day_visit*), Week mode visit (*Week_visit*) and Number of visits (*Visit_num*) are higher than that of others in both



(a) Honolulu (b) London

FIGURE 11. Average Percentile Rank (APR) comparison results for the single feature.

Honolulu and London dataset and the value of Number of visitors (*Visitor_num*) is the lowest in both datasets.

In conclusion, the tracking approach proposed in this paper performs better than the baseline methods. It can produce the next location list which contains the true next place with a relatively high possibility and rank it in the front position of the list.

VII. CONCLUSION

In this paper, we propose an efficient target tracking approach through mobile crowdsensing to solve the target tracking problem. The mobile users communicate with the MCS platform only when they find the appearance of the target. Hence, the mobile users form a crowdsensing network by means of wireless communications, which consume low energy resources. By collecting the reports of mobile users, the approach builds a location history list of the target, so that the searcher could track the target according to the location history list. Sometimes the searcher probably fails to find the target due to the mobility of the target. To overcome this problem, we propose and formulate a next place prediction problem to predict the next place list that may be visited by the target and address this problem as a binary classification problem. Furthermore, we utilize a location prediction model, named XGBoost, to predict the next places possibly to be visited by the target. Specifically, for each place in the candidate place set, we predict the probability that will be visited by the target and build a next place list which contains top-k places that the target will most likely reach. To improve the prediction performance of the model, we extract the combined features including individual mobility feature, global mobility feature as well as context feature to train the prediction model. Finally, we evaluate the model on a large-scale real-world dataset. The experimental results demonstrate that the proposed approach can effectively track target movement.

REFERENCES

[1] J. Li, Y. Huang, Y. Wei, S. Lv, Z. Liu, C. Dong, and W. Lou, "Searchable symmetric encryption with forward search privacy," *IEEE Trans. Depend. Sec. Comput.*, to be published.

[2] Z. Liu, B. Li, Y. Huang, J. Li, Y. Xiang, and W. Pedrycz, "NewMCOS: Towards a practical multi-cloud oblivious storage scheme," *IEEE Trans. Knowl. Data Eng.*, to be published.

[3] K. Banti, M. Louta, and G. Karetsos, "ParkCar: A smart roadside parking application exploiting the mobile crowdsensing paradigm," in *Proc. Int. Conf. Inf., Intell., Syst. Appl.*, Aug. 2017, pp. 1–6.

[4] B. Piao and K. Aihara, "Detecting the road surface condition by using mobile crowdsensing with drive recorder," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[5] M. Marjanović, S. Grubeša, and I. P. Žarko, "Air and noise pollution monitoring in the city of Zagreb by using mobile crowdsensing," in *Proc. Int. Conf. Softw., Telecommun. Comput. Netw.*, Sep. 2017, pp. 1–5.

[6] A. Farshad, M. K. Marina, and F. Garcia, "Urban WiFi characterization via mobile crowdsensing," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[7] Z. Peng, S. Gao, B. Xiao, S. Guo, and Y. Yang, "CrowdGIS: Updating digital maps via mobile crowdsensing," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 369–380, Jan. 2017.

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

[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[10] L. Liu, X. Zhang, and H. Ma, "Dynamic node collaboration for mobile target tracking in wireless camera sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1188–1196.

[11] H. Ma, M. Yang, D. Li, Y. Hong, and W. Chen, "Minimum camera barrier coverage in wireless camera sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 217–225.

[12] H. Shin, T. Park, B. Lee, B. Lee, J. Song, Y. Chon, and H. Cha, "Cosmic: Designing a mobile crowd-sourced collaborative application to find a missing child *in situ*," in *Proc. Int. Conf. Hum.-Comput. Interact. Mobile Devices Services*, Sep. 2014, pp. 389–398.

[13] Y. Jing, B. Guo, Z. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Crowd-Tracker: Optimized urban moving object tracking using mobile crowd sensing," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3452–3463, Oct. 2017.

[14] N. Cao, S. Brahma, and P. K. Varshney, "Target tracking via crowdsourcing: A mechanism design approach," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1464–1476, Mar. 2015.

[15] Y.-J. Kim and S.-B. Cho, "A HMM-based location prediction framework with location recognizer combining k-nearest neighbor and multiple decision trees," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, 2013, pp. 618–628.

[16] Y. Qiao, Z. Si, Y. Zhang, F. B. Abdesslem, X. Zhang, J. Yang, Y. Qiao, Z. Si, Y. Zhang, and F. B. Abdesslem, "A hybrid Markov-based model for human mobility prediction," *Neurocomputing*, vol. 278, pp. 99–109, Feb. 2017.

[17] J. A. Alvarez-Garcia, J. A. Ortega, L. Gonzalez-Abril, and F. Velasco, "Trip destination prediction based on past GPS log using a Hidden Markov Model," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8166–8171, 2010.

[18] S. Parija, R. K. Ranjan, and P. K. Sahu, "Location prediction of mobility management using neural network techniques in cellular network," in *Proc. Int. Conf. Emerg. Trends VLSI, Embedded Syst., Nano Electron.*, Jan. 2013, pp. 1–4.

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

[20] S. M. Rahimi and X. Wang, "Location recommendation based on periodicity of human activities and location categories," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, Apr. 2013, pp. 377–389.

[21] 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 (CBD)*, Aug. 2017, pp. 238–243.

[22] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Proc. IEEE Int. Conf. Data Mining*, Jan. 2013, pp. 1038–1043.

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

[24] Y. Song, Z. Hu, X. Leng, H. Tian, K. Yang, and X. Ke, "Friendship influence on mobile behavior of location based social network users," *J. Commun. Netw.*, vol. 17, no. 2, pp. 126–132, Apr. 2015.

[25] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Diego, Ca, USA, Aug. 2011, pp. 1082–1090.

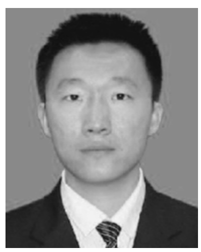
[26] C. Yu, Y. Liu, D. Yao, L. T. Yang, H. Jin, H. Chen, and Q. Ding, "Modeling user activity patterns for next-place prediction," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1060–1071, Jun. 2017.

[27] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proc. ACM Int. Conf. Conf. Inf. Knowl. Manage.*, Nov. 2014, pp. 739–748.

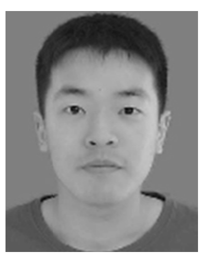
[28] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 263–272.

DONGMING LUAN received the B.E. degree in software engineering from Jilin University, Changchun, Jilin, China, in 2017, where he is currently pursuing the master's degree with the College of Computer Science and Technology. His current research interest includes mobile crowdsensing.

YONGJIAN YANG received the B.E. degree in automation from the Jilin University of Technology, Changchun, Jilin, China, in 1983, the M.E. degree in computer communication from the Beijing University of Posts and Telecommunications, Beijing, China, in 1991, and the Ph.D. degree in software and theory of computer from Jilin University, in 2005, where he is currently a Professor and a Ph.D. Supervisor, the Vice Dean of the Software College of Jilin University, the Director of Key Laboratory under the Ministry of Information Industry, the Standing Director of the Communication Academy, and a member of the Computer Science Academy of Jilin Province. His research interests include network intelligence management, wireless mobile communication and services, and wireless mobile communication.



EN WANG received the B.E. degree in software engineering from Jilin University, Changchun, in 2011, and the M.E. degree in computer science and technology and the Ph.D. degree in computer science and technology from Jilin University, Changchun, in 2013 and 2016, respectively, where he is currently a Lecturer with the Department of Computer Science and Technology. He is also a Visiting Scholar with the Department of Computer and Information Sciences, Temple University, Philadelphia. His current research interests include the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowdsensing.



QIYANG ZENG received the B.E. degree in software engineering from Jilin University, Changchun, in 2017, where he is currently pursuing the master's degree in software engineering. His current research interest includes machine learning.

ZHAOHUI LI received the Ph.D. degree in control theory and control engineering from Nankai University, in 2003, where he is currently with the College of Cyber science. His research interest includes information security.

LI ZHOU received the Ph.D. degree in management from the Beijing Institute of Technology, Beijing, China, in 2006. She is currently a Professor and the Dean of the School of Information, Beijing Wuzi University. She is also the Director of the Beijing Intelligent Logistics System Cooperative Innovation Center. Her research interests include intelligent logistics systems, optimization theory and methods, and big data technology and applications.