Trajectory Penetration Characterization for Efficient Vehicle Selection in HD Map Crowdsourcing

Xiaofeng Cao, *Graduate Student Member, IEEE*, Peng Yang, *Member, IEEE*, Feng Lyu, *Member, IEEE*, Jiarong Han, Yan Li, Deke Guo, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—In this article, we investigate the worker (i.e., vehicle) selection problem in vehicle-based crowdsourcing (VBC), where vehicles in a specific area are recruited by the crowdsourcing platform to collect geographical information in real driving scenarios for autonomous driving. Given a limited recruitment budget, we formulate a cumulative platform utility maximization problem (CMP) to obtain the optimal worker set. The CMP is unsolvable directly as the platform has no prior information of workers at the initial stage (also known as "cold start") and the cost of collecting all workers' information is prohibitive. To solve the problem, we first conduct a comprehensive data analytics on two real-world vehicle traces and obtain two crucial observations: 1) trajectory of individual vehicle is highly uncertain that it is difficult to make accurate prediction and 2) the overall distribution of vehicular trajectory penetration (measured by collection quantity and coverage) has a diurnal pattern and varies with weekly periodicity. Inspired by the insights, we propose the performance transfer-based online worker selection (*POSE*) scheme, which works independently from trajectory prediction with two components, i.e., transfer learning-based performance estimation and online worker selection (OWS). Based on the diurnal pattern, the former component collects a short-period trajectory penetration data of vehicles for model fitting, which can output a specific numerical distribution. With the fitting model, we can identify and select vehicles with high trajectory penetration at the initial stage to cope with the "cold start" problem. Then, we map the worker selection problem into a multiarmed bandit problem and develop upper confidence bound-based approach to solve it. Extensive trace-driven simulations are carried out and the results demonstrate the efficiency of *POSE* in terms of cumulative platform utility.**

***Index Terms*—High-definition (HD) map collection, Internet of Things, transfer learning, vehicle-based crowdsourcing (VBC), vehicular trajectory penetration, worker selection.**

Manuscript received June 22, 2020; revised September 2, 2020; accepted September 22, 2020. Date of publication September 30, 2020; date of current version March 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62002389 and Grant U19B2024, and in part by the Natural Sciences and Engineering Research Council of Canada. This article was presented in part at the IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, Dec. 2019. *(Corresponding author: Xiaofeng Cao.)*

Xiaofeng Cao, Yan Li, and Deke Guo are with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: caoxiaofeng10@nudt.edu.cn; liyan10@nudt.edu.cn; dekeguo@nudt.edu.cn).

Peng Yang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yangpeng@hust.edu.cn).

Feng Lyu is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: fenglyu@csu.edu.cn).

Jiarong Han is with the School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: hanjiarong@bupt.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2020.3028026

I. Introduction

AUTONOMOUS vehicles (AVs) with various levels of automation are already on the road [2]–[5]. It is expected that AVs with full automation will provide safer and more efficient transportation [6], [7]. A prerequisite for AVs is to understand the driving environment in real time. Generally, AVs rely on a combination of the premade environment map [e.g., high-definition (HD) map] and onboard sensors (e.g., LIDAR, GPS, camera, and accelerometer) to sense the surrounding environment and locate themselves relatively in the map, based on which online driving decisions can be made. Therefore, the environment map is of paramount importance and directly determines the safety and quality of driving experience [2], [8]. For map construction and update, vehicle-based crowdsourcing (VBC) is a promising solution with low cost and high efficiency [9]. Particularly, with various onboard sensors and similar working scenarios, vehicles are able to collect rich environmental information in real driving scenarios [10]. Generally, an urban VBC scenario consists of a local platform, workers (i.e., vehicles),[1] and tasks. The platform is supported by edge nodes within an urban area [11], [12], which publishes tasks to local workers and selects workers from the candidates, who are willing to participate in the crowdsourcing. Then, edge nodes collect and aggregate information from selected workers and send the processed map information to the platform to complete the tasks.

Worker selection is the primary factor affecting crowdsourcing quality. The crowdsourcing platform makes selection decisions based on various observations, such as the vehicle mobility and historical performance of completing crowdsourcing tasks [13], [14], and gives workers either monetary or nonmonetary incentives [12], [15]. The objective is to maximize the crowdsourcing quality with as less worker recruitment cost as possible. However, for the map collection, selecting a suitable subset of workers from all candidates can be very challenging due to the following reasons. First, the tasks of spatial crowdsourcing are usually scattered in different geographical locations. In contrast, the HD map collection

[1]In this article, vehicles are the workers in the VBC scenario. Hence, "vehicle" and "worker" are interchangeable in the following sections.

requires continuous and complete coverage of the target area, posing extra continuity constraints on map collection tasks. Therefore, to achieve an efficient worker selection, the platform calls for fine-grained information of vehicle trajectory. Second, the spatial–temporal information of vehicle trajectory changes rapidly. Different vehicles also have diverse and unpredictable driving routes, which make the selection more complicated. Third, under the budget constraint, maximizing the crowdsourcing utility (i.e., collection quantity and collection coverage) and minimizing the platform cost are competing objectives, it is a challenging task to achieve a satisfying tradeoff between them.

On the other hand, the platform usually has no historical performance information of workers at the initial stage of crowdsourcing, i.e., facing the "cold start" problem [16]. As a result, the platform has to experience a long period to aggregate the worker data, during which the system may suffer from a low-efficiency crowdsourcing process. One plausible approach to solve the "cold start" problem is to select all the vehicles and collect the corresponding data in a short period. However, it is quite difficult as the vehicle set in an urban area is large and time varying, and thus, the time and cost of data collection are prohibitive.

To cope with the above challenges, we investigate the worker selection of VBC and formulate a cumulative platform utility maximization problem (CMP) to maximize the long-term VBC utility, taking both the collection quantity and coverage into account. To solve this problem, we propose a systematic framework, which consists of data analytics and strategy design. In the data analytics, we divide the target urban area into small square blocks and analyze two real-world vehicular traces[2] in San Francisco and Rome. We then achieve several valuable insights that can benefit the scheme design.

1) The distribution of daily trajectory penetration (measured by collection quantity and coverage) of vehicles has an exponential decay, indicating there exists a small portion of vehicles with relatively high trajectory penetration. By recruiting them, it is possible to achieve high-quality VBC with a small budget cost.
2) The trajectory of an individual vehicle is highly uncertain and it is hard to predict the vehicle trajectory with high accuracy. To be immune from this restriction, the underlying worker selection approach should work independently from the trajectory prediction.
3) The overall distribution of vehicular trajectory penetration has a diurnal pattern and varies on weekly periodicity, which means the future long-term vehicle driving activities can be estimated based on short-time learning. Such a feature can be harnessed to develop a transfer learning-based method to solve the "cold start" problem.

In this article, we propose a worker selection scheme named *POSE*, i.e., performance transfer-based OWS, which consists of two components: 1) the transfer learning-based performance estimation and 2) OWS. In the former component, relying on the weekly periodicity, we collect the trajectory penetration data of a small number of vehicles in one week. By performing data fitting, we find that the trajectory penetration data approximately follow a specific distribution at each temporal context. With the fitting model, we can estimate the performance of the newly joined vehicles (i.e., the vehicles without any historical trajectory penetration information) by transferring the trajectory penetration knowledge of previously selected vehicles with the same temporal feature. Furthermore, we theoretically prove that at the initial stage of VBC, the transfer learning-based performance estimation can achieve high-quality crowdsourcing with fast data aggregation.

With the aggregated data at the initial stage, the worker selection problem then can be mapped to a real-time worker exploration and exploitation (E2) process [19], [20] and modeled as a multiarmed bandit problem. As E2 integrates the information collection (exploration) and decision making (exploitation), we can continuously strike the balance between exploring possible better decisions and exploiting current optimal decisions. It can provide a workable online solution without any prior assumptions. Finally, we devise the OWS algorithm to solve the CMP with the time complexity of $\mathcal{O}(W \log_2 W)$. In OWS, we employ the well-known E2 algorithm [i.e., the upper confidence bound (UCB) algorithm] to solve the multiarmed bandit problem for map collection. Essentially, OWS always opportunistically chooses the workers with the highest upper bound of worker utility. Moreover, we present a guideline on efficiently balancing the E2 during OWS. The experimental results demonstrate that the proposed scheme can significantly improve the cumulative platform utility during both the initial stage and the overall crowdsourcing period.

We highlight our main contributions in this article as follows.

1) We consider the HD map collection in urban areas, which is an essential building block for autonomous driving. Specifically, we adopt the edge-enabled VBC to collect and update the map data and formulate the CMP to obtain the optimized worker set.
2) We propose a systematic framework to solve the CMP. First, two urban-scale GPS traces are analyzed to characterize vehicular trajectory penetration in terms of distribution, correlation, and variation. Based on these insights, we design and implement a worker selection scheme called *POSE*, which consists of two components: a) the transfer learning-based performance estimation and b) OWS. They can efficiently address the "cold start" problem and the OWS problem, respectively.
3) Extensive experiments demonstrate the efficacy of *POSE*. It is worth noting that *POSE* can be readily extended to other urban VBC applications (e.g., traffic monitoring or road anomaly detection) with two exemplary merits: a) the platform can customize the worker selection in accordance with the planned budget and b) the real urban-scale GPS trace is adopted, which can provide a solid performance reference for implementations.

[2]With the availability and less privacy concern, the taxi traces are widely used in the research area of VBC [14], [17], [18].

The remainder of this article is organized as follows. Section II presents the related work. Section III details the system model and problem formulation. In Section IV, we give the data description and conduct comprehensive data analytics on trajectory penetration of vehicles. Section V elaborates our *POSE* design. In Section VI, we evaluate the performance of *POSE* by trace-driven simulations. Finally, we conclude this article and direct our future work in Section VII.

II. Related Work

Spatial crowdsourcing is an efficient paradigm for geographical data collection, which requires each worker to perform the task physically, where the spatiotemporal information, such as location, time, mobility, and the associated context, plays critical roles [21], [22]. In particular, the VBC is regarded as a promising solution for urban-scale crowdsourcing due to its low cost and high efficiency. As vehicles have high mobility and are equipped with rich onboard sensors, they can achieve larger coverage and collect various environmental information. Many research works on VBC have been developed recently for various purposes, e.g., traffic monitoring [23], air quality monitoring [24], pavement condition testing [25], and roadway network reconstruction [26].

Worker selection is one of the important problems in VBC. However, the high mobility of vehicles makes this problem complicated and challenging. Researchers consider vehicles' mobility based on randomly generated or model-based trajectories. To maximize the coverage with minimum cost, Abdelhamid *et al.* [27] proposed a framework consisting of a reputation assessment scheme, a pricing model, and a selection scheme for worker recruitment, where the vehicular trajectories are randomly generated within a 5-km roadway. With the budget constraint, Hu *et al.* [28] designed a duration-variable principle in the participant recruitment problem, aiming at maximizing the collecting coverage. The trajectory of vehicles follows a probabilistic model. Similarly, Wang *et al.* [13] studied two trajectory models of the vehicles: deterministic and probabilistic model, and the worker selection problem based on both models is proved to be NP-hard. Then, the problem under different mobility models is solved based on heuristic algorithms, where the approximation ratio to the optimum is given. Yi *et al.* [14] also investigated the Gauss–Markov mobility model in the worker selection problem and proposed a fast recruitment algorithm. These works, while efficient in worker selection or recruitment, are based on strong assumptions of the vehicles' mobility model.

Other works assume that the trajectory of vehicles can be accurately predicted, and thus the future trajectories can be known ahead. Zhang *et al.* [10] formulated the worker recruitment problem as a biobjective (i.e., for the location-based query tasks and automatic sensing tasks) optimization problem, which is proved to be NP-hard. To address this problem, they propose two heuristic algorithms, both of which require the vehicle trajectory information. Similarly, based on the predicted trajectories, He *et al.* [29] presented a new participant recruitment strategy for VBC, which guarantees that the system can perform well using the currently recruited participants for a period of time in the future. For the incentive mechanism design of worker selection, both Restuccia *et al.* [17] and Gao *et al.* [30] assumed that future trajectories are known in advance and evaluated their algorithms over large-scale taxi GPS data trace. Likewise, Xu *et al.* [31] predicted the trajectories of vehicles and the number of task requests with the Markov model and the time-space graphical model, respectively. Then, the iLOCuS scheme is presented to incentivize vehicles to adjust their trajectories to the requested tasks, therefore completing tasks with minimum cost. However, it is difficult to capture and predict the real-world vehicles' trajectory accurately, since individual vehicle's trajectory in the urban area is dynamic and highly uncertain. Although some deep learning methods have been proposed in the prediction of the city-wide traffic, there is still a gap to apply these methods in urban vehicle trajectory prediction, and the cost can be prohibitive in both time and expenditure. Moreover, the "cold start" problem is rarely considered in these works, as they assume that the vehicle's trajectory is already known or easy to get.

In our previous work [1], we have demonstrated the efficacy of combining data analytics and the UCB-based approach in OWS. In addition to our preliminary work, this article makes the following improvements: 1) we propose the *POSE* scheme, which consists of transfer learning-based performance estimation and OWS. It can efficiently solve the "cold start" problem at the initial stage and achieve superior VBC performance; 2) we define the vehicle trajectory penetration to measure the collection quantity and coverage in VBC. Then, we conduct comprehensive data analytics to characterize the penetration of vehicles in terms of distribution, correlation, and variation, which inspire our scheme design; and 3) we theoretically prove the performance of the knowledge transfer and integrate it into the OWS algorithm.

III. System Overview and Problem Formulation

In this section, we first describe the crowdsourced HD map collection in the urban area and then formulate the VBC worker selection problem.

A. Crowdsourced Map Collection in Urban Area

We consider an urban area and divide it into small areas. The map collection of the target urban area is the crowdsourcing task, and each small area corresponds to a microtask. The VBC system consists of two entities: 1) the platform and 2) the worker. Workers are a group of vehicles who contribute to the crowdsourcing task and receive incentives from the platform. The platform consists of a central cloud node and multiple edge nodes (e.g., base stations). It broadcasts the spatial crowdsourcing task through edge nodes to vehicles and selects a subset of them to perform the tasks, aiming to achieve the maximum crowdsourcing utility. Then, edge nodes collect and aggregate all the collected information from selected vehicles, and send the processed map information to the platform. Note that the platform has a limited recruitment budget to employ vehicles.

B. System Model

Supposing the time span is divided into a sequence of time slots $\mathcal{T} = \{1, 2, \ldots, t, \ldots\}$, the map collection

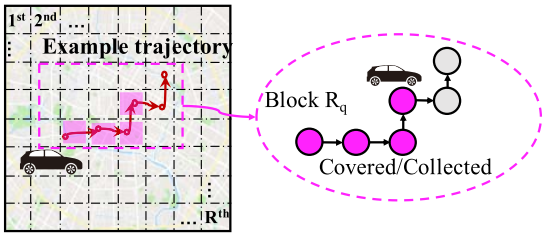

Fig. 1. Overview of the HD map collection by VBC.

takes a fixed period, and there are varying sets of workers that can be recruited at different time slots. Let $\mathcal{W}_t = \{w_{t,1}, w_{t,2}, \ldots, w_{t,m}, \ldots\}$ be the worker candidate set at time slot t, where m is the worker index. As shown in Fig. 1, without loss of generality, we consider a rectangular urban area, which is partitioned into small square blocks.[3]

Definition 1 (Crowdsourcing Tasks): Our VBC task is the map collection of the target area, which can be divided into microtasks. Each microtask corresponds to a specific square block, and all microtasks collectively provide full information of the target area. When the vehicle enters the block, part of the corresponding block map will be collected, as indicated by the pink square on the left-hand side and pink circle on the right-hand side of Fig. 1. As the map changes over time (i.e., the map of each block is different at each time slot), it should be updated continuously across the whole time span.

Definition 2 (Worker Trajectory and Daily Collection): The worker trajectory is a set of connected blocks sequentially covered by the worker. We denote $\mathcal{R} = \{r_1, r_2, \ldots, r_q\}$ as a set of q target blocks. Then, from the worker trajectory, we can obtain the worker daily collection. Particularly, the daily collection of worker $w_{t,m}$ at time slot t can be defined as a vector of $\boldsymbol{s}_{t,m} = \{s_{t,m,q}\}_{r_q \in \mathcal{R}}$, where $s_{t,m,q}$ is the quantity that the worker $w_{t,m}$ has covered the block r_q during the time slot t. If the block r_q has not been covered during the time slot t, then we have $s_{t,m,q} = 0$.

We assume blocks are with different importance, and the weights of blocks are given by a vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_q)$, which can be determined in advance. The weight of each block can be affected by factors, such as the geographic location, traffic condition, etc. For example, the map of a block with a larger number of crossings (e.g., a transportation hub) has a higher weight than others. The repeated coverage of a certain block by multiple vehicles impairs the system efficiency. To measure such spatiotemporal redundancy, we introduce the redundancy degree, which is defined as a vector $\boldsymbol{p}_t = (p_{t,1}, p_{t,2}, \ldots, p_{t,q})$, where $p_{t,q}$ is the redundancy degree of block r_q at time slot t and calculated by

$$p_{t,q} = \frac{1}{l_{t,q}} \tag{1}$$

where $l_{t,q}$ means the quantity that block r_q has been covered in time slot t. This equation indicates that a lower value of redundancy degree implies a high redundancy of a block, and thus limits the worker's contribution to the platform.

[3]For ease of illustration, we divide the target area into square regions. According to different service scenarios, the size of the block could be customized flexibly.

Definition 3 (Worker Utility): worker utility $\pi_{t,m}$ is the benefit that a single worker $w_{t,m}$ can bring to the platform at time slot t. It can be calculated by the daily collection of worker $w_{t,m}$ during the time slot t and the corresponding blocks' redundancy degree $\boldsymbol{p}_t$. Formally, it can be defined as:

$$\pi_{t,m} = \boldsymbol{s}_{t,m}(\boldsymbol{v} \odot \boldsymbol{p}_t)^T \tag{2}$$

where $\odot$ is the Hadamard product between two vectors. From (2), we know that the worker utility is determined by the collection quantity (related to $\sum_{r_q \in \mathcal{R}} s_{t,m,q}$) and the collection coverage (related to the number of $s_{t,m,q} > 0$). The worker with high *trajectory penetration* (i.e., high collection quantity and high collection coverage) can contribute more to the platform.

Definition 4 (Platform Utility): Platform utility is the total utility of the selected worker set $\mathcal{W}'_t$ at the time slot t. Formally, it can be defined as

$$P_t = \sum_{w_{t,m} \in \mathcal{W}'_t} \pi_{t,m} \tag{3}$$

where $\mathcal{W}'_t \subseteq \mathcal{W}_t$.

Definition 5 (Crowdsourcing Budget): The platform needs to incentivize workers to contribute to the map collection tasks. The cost of selecting a worker $w_{t,m}$ varies at different time slots, and we use $\mathcal{C}_t = \{c_{t,1}, c_{t,2}, \ldots, c_{t,m}, \ldots\}$ to represent the cost set at each time slot, where $c_{t,m}$ is the cost of selecting worker $w_{t,m}$ at time slot t. The cost of recruiting a worker in a specific time slot can be acquired through existing approaches such as online bidding [17]. Under the recruitment budget constrain, the total cost of recruiting workers at each time slot is less or equal than K. Moreover, in order to suppress tricky behaviors from strategic workers, only the data collected in the current time slot is valid. Besides, we measure the redundancy degree both temporally and spatially, thus the strategic workers (e.g., always back and forth with a similar trajectory) will be identified, and the probability of selecting these workers will be significantly reduced. The blockchain technology can also be a possible solution to reliable crowdsourcing, which enables a truthful and decentralized mechanism [32].

C. Problem Formulation

With the above system model, we formulate the worker selection problem as the CMP.

Given: Candidate worker sets $\mathcal{W}_t$, and corresponding costs $\mathcal{C}_t$, interested blocks set $\mathcal{R}$, and the worker daily collection $\boldsymbol{s}_{t,m}$.

Objective: Recruiting a subset of workers $\mathcal{W}'_t$ at time slot t such that: $\sum_{\mathcal{W}'_t \subseteq \mathcal{W}_t, t \in \mathcal{T}} P_t$ is maximized.

Subject to: The budget in each time slot, $\sum_{w_{t,m} \in \mathcal{W}'_t} c_{t,m} \leq K \ \forall t \in \mathcal{T}$.

It is difficult to solve the CMP problem directly due to the following reasons. First, the trajectories of vehicles are time varying and unavailable in advance, and the cost of collecting all vehicles' trajectory penetration information is prohibitive in urban areas. Meanwhile, there is no well-established model that can accurately capture the uncertainty and dynamics of the vehicle's trajectory within a long period. Moreover, even all

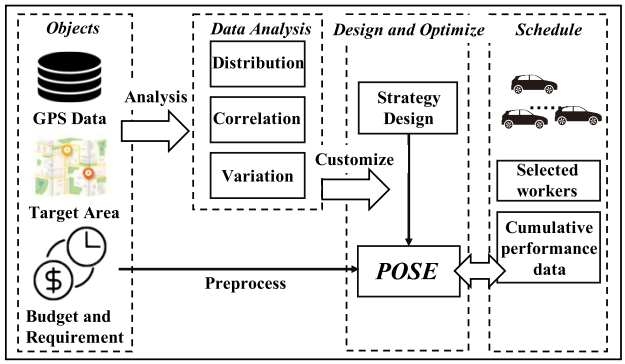

Fig. 2. Systematical framework.

future information is known, the above maximization problem with cardinality constraint has been proved to be NP-hard [29] with time complexity $\mathcal{O}(W2^W)$. Solving this problem is time consuming, especially in the urban scenario.

In what follows, we solve this problem based on a systematic framework shown in Fig. 2. First, the target area, the GPS data, and the budget are taken as input. Then, we perform data analytics and reveal the underlying characteristics (e.g., the distribution, correlation, and variation of trajectory penetration) of vehicles in urban areas. Based on the above insights, on the platform, we design and implement the *POSE* scheme for worker selection.

IV. Data Preprocessing and Analysis

In this section, we give the data description and details of preprocessing. Then, we conduct the overall trajectory penetration analysis from both the temporal and spatial dimensions, to figure out crucial factors that affect worker selection performance.

A. Data Description and Preprocessing

To understand the realistic traffic statistics and design an informed worker selection strategy, the GPS trace of taxis can be useful for two reasons. First, taxis are usually more active and they cover a wide range of urban areas, which means that taxis have high potential in VBC. Analyzing GPS trace of taxis can directly benefit the future VBC application. Second, taxis are public transportation and widely considered in VBC tasks [12], as there are fewer privacy concerns compared with the private vehicles. Hence, we analyze two urban GPS taxi traces in Rome [33] and San Francisco [34]. These real mobility traces are generated by the onboard GPS-enabled devices, with detailed information as follows.

1) *San Francisco:* The trace collected in San Francisco contains the GPS data of 536 taxis, with a time span of 25 days. Each taxi was equipped with a GPS receiver and periodically (every minute) sent the location information to the central server, including the vehicle id, timestamp, latitude, and longitude. There are totally 11 219 955 GPS records of San Francisco taxi trace.
2) *Rome:* The trace data set collected in Rome contains the GPS data of 316 taxis, with a time span of 30 days. Each taxi was equipped with a GPS receiver and periodically (every 7 s on average) sent the location information to the central server. The positions with a precision error higher than 20 m were ignored. There are in total 21 817 851 GPS records.

TABLE I
Main Statistics of Data Sets

Trace	San Francisco	Rome
Duration	25 days	30 days
Total number of vehicles	536	316
Average sampling rate	60 seconds	7 seconds
Total number of records	11,219,955	21,817,851

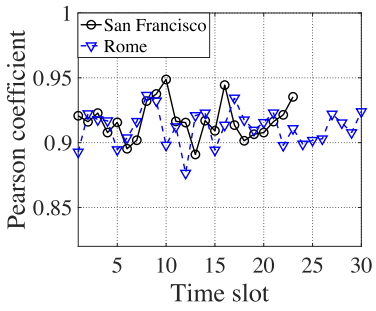

Fig. 3. Pearson coefficient.

Key statistics of these data sets are presented in Table I. First, we find two main kinds of errant data that should be dealt with: 1) positioning drift: the GPS data show that vehicles' GPS traces are not right on the road segments and 2) missing data: some carpool vehicles' GPS data are not uploaded within a given time period. For example, the GPS data should be reported every 2–4 s, but no data are reported within 60 s in some time periods. To address the above errors, we relocate those off-road data points to the nearest road segments and divide the target urban areas of two cities into 100 blocks.[4] Then, we convert the original GPS record of taxis into the trajectory record. Since consecutive GPS records can be regarded as a trajectory, we set a time threshold (e.g., 100 s), where GPS records with time differences larger than 100 s belong to different trajectories. In a trajectory, if a taxi has several consecutive GPS records located in the same block, only the last record information will be used. For example, as shown in Table II, the time difference between GPS records 5 and 6 is over 100 s, thus the GPS records are divided into two trajectories. In trajectory 1, the GPS records 1 and 2 are located in the same block (i.e., block 15), and the time of the GPS record 2 will be used. Thus, trajectory 1 is (15015844, block 15) $\rightarrow$ (15015860, block 25) $\rightarrow$ (15015870, block 26) $\rightarrow$ (15015878, block 36). In this way, we convert all the GPS records to trajectories.

B. Correlation Between Collection Quantity and Coverage

First, we check the correlation between the vehicle collection quantity and coverage. We represent the vehicle collection quantity of worker candidate $w_{t,m}$ at time slot t as v_t^m, which can be calculated by $v_t^m = \sum_{r_q \in \mathcal{R}} s_{t,m,q}$. The vehicle

[4]The block size is mainly determined by the topological structure of urban roads, where ensuring that there are roads in each grid.

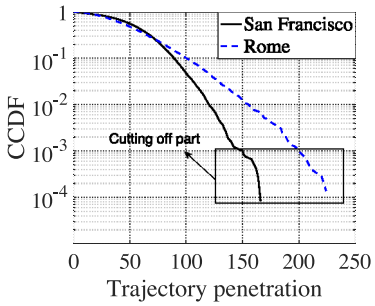

Fig. 4. CCDFs.

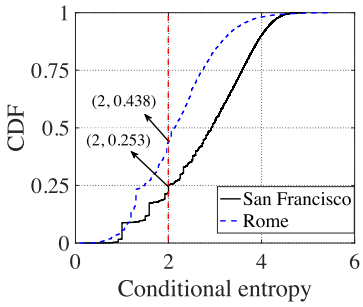

Fig. 5. Conditional entropy.

TABLE II
EXAMPLE OF THE TRAJECTORY EXTRACTION

GPS Record #	Longitude	Latitude	Time (s)	Block #	Trajectory #
1	12.49413	41.90140	15015840	15	1
2	12.49414	41.90145	15015844		
3	12.49434	41.90176	15015860	25	
4	12.49440	41.90180	15015870	26	
5	12.49452	41.90190	15015878	36	
6	12.49450	41.90210	15015986	37	2
7	12.49451	41.90230	15015990	47	
8	12.49453	41.90240	15015993		
9	12.49449	41.90238	15015997		
10	12.49459	41.90250	15016001	48	

collection coverage can be measured by vehicle's entropy[5] H_t^m, which can be interpreted as the diversity of blocks that the vehicle $w_{t,m}$ covered at time slot t. Then, we take pearson coefficient [35] to measure the correlation, and the *daily pearson coefficient* of vehicles can be calculated as

$$\rho^t = \frac{\sum_{w_{t,m}\in\mathcal{W}_t}\left(H_t^m - \overline{H_t^m}\right)\left(v_t^m - \overline{v_t^m}\right)}{\sqrt{\sum_{w_{t,m}\in\mathcal{W}_t}\left(H_t^m - \overline{H_t^m}\right)^2}\sqrt{\sum_{w_{t,m}\in\mathcal{W}_t}\left(v_t^m - \overline{v_t^m}\right)^2}} \tag{4}$$

where $\overline{H_t^m} = [(\sum_{n\in\mathcal{T}} H_t^m)/|\mathcal{T}|]$ and $\overline{v_t^m} = [(\sum_{n\in\mathcal{T}} v_t^m)/|\mathcal{T}|]$. The results are plotted in Fig. 3, from which we can see that the pearson coefficient is large in both cities. The pearson coefficient fluctuates from 0.89 to 0.95 in San Francisco and fluctuates from 0.88 to 0.94 in Rome. We can infer that the vehicle collection quantity and coverage have a strong correlation, indicating that the vehicle with a larger collection quantity is usually with a better collection coverage. Therefore, in what follows, we will adopt the value of the collection quantity as the driving factor of trajectory penetration to cast our worker selection strategy.

C. Exponential Distribution of Vehicle Trajectory Penetration

We then examine the complementary cumulative distribution function (CCDF) of daily trajectory penetration (i.e., collection quantity) of vehicles. Fig. 4 shows the CCDF of daily trajectory penetration of vehicles in two cities. We can see that *the distribution of trajectory penetration has an exponential decay* in both cities since the clear linear plot appears under the linear-log scale of the CCDF result [36]. Then, we observe that there is a cutting-off part at the tail of the daily penetration distribution. Both observations indicate that the vehicle with relatively large trajectory penetration takes a small portion, which is crucial to our system design. By recruiting these vehicles, it is possible to achieve high-quality VBC with a small budget cost.

[5]Since the similar calculation will be presented in the later definition of conditional entropy, the calculation of the vehicle daily entropy is omitted.

D. High Dynamics of Individual Trajectory

In this section, we investigate vehicle penetration variation from both temporal and spatial aspects. First, we define *conditional entropy* to indicate the vehicle trajectory uncertainty. In particular, based on the historical trajectories, we can get the vehicles' transition probability matrix between blocks. The matrix can be written as

$$\boldsymbol{a} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,j} & \cdots & a_{1,q} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \cdots & a_{i,j} & \cdots & a_{i,q} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{q,1} & \cdots & a_{q,j} & \cdots & a_{q,q} \end{bmatrix} \tag{5}$$

where $a_{i,j}$ represents the probability that vehicles transit from block r_i to block r_j. The sum of each row equals 1, i.e., $\sum_{j=1}^{q} a_{i,j} = 1$. Note that $q = 100$. Therefore, the conditional entropy of block r_i can be defined as

$$H_{r_i} = \sum_{j=1}^{q} a_{i,j} \log_2 \frac{1}{a_{i,j}} \tag{6}$$

where a larger entropy indicates more uncertainties of the trajectory traversing this block.

Fig. 5 shows the cumulative distribution function (CDF) of the conditional entropy of all blocks across the whole time span in San Francisco and Rome, respectively. It can be observed that 74.7% and 56.2% of the conditional entropy are greater than 2 in San Francisco and Rome, respectively. In this case, the trajectory is highly dynamic in both cities, and thus it is difficult to achieve future trajectory prediction with high accuracy. We further check the variation of the vehicle daily penetration by the *penetration changing ratio*, which can be calculated as

$$\varphi_{t,m} = \frac{v_{t,m} - v_{t-1,m}}{v_{t-1,m}} \tag{7}$$

where a large changing ratio means the trajectory penetration has a significant fluctuation and is not stable over the

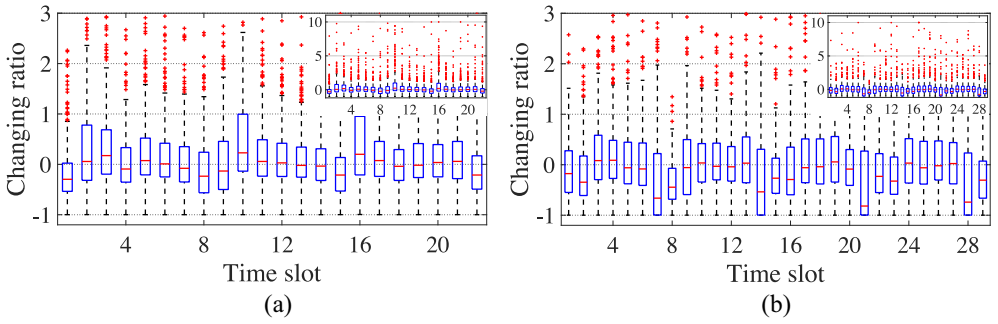

Fig. 6. Changing ratio of the vehicle trajectory penetration. (a) San Francisco. (b) Rome.

temporal dimension. Fig. 6 presents the boxplot of the trajectory penetration changing ratio in both cities. We can find that vehicles have a wide range of changing ratio, with the minimum -0.99 and maximum 9.97 in San Francisco, and the minimum -0.99 and maximum 9.95 in Rome. The changing ratio also remains high throughout the entire time span and has a considerable overall average value, which is 0.57 in San Francisco and 0.69 in Rome. Those figures demonstrate that not only the trajectory has high uncertainty but also the vehicle trajectory penetration varies dynamically. The trajectory prediction-based approach has been applied to many crowdsourcing applications in existing works, with which the platform can obtain the vehicle trajectory in advance and make the worker selection accordingly. We recognize that this may be an effective method for small-scale crowdsourcing applications. However, due to the dynamics and uncertainties, this can be time consuming and may not work for the massive crowdsourcing tasks in large-scale urban areas.

In summary, the underlying mobility pattern verifies that: 1) it is possible to achieve high-quality crowdsourcing by selecting a small number of vehicles; 2) the trajectories of vehicles are uncertain and highly dynamic; and 3) the vehicle collection quantity and geographical coverage have a strong correlation, and we will take collection quantity as the driving factor of vehicle trajectory penetration. To this end, given the limited budget, we plan to design a worker selection strategy, which is independent of the trajectory prediction.

V. Design of POSE

In this section, we elaborate on the design of *POSE*, which runs with two stages: 1) the initial stage and 2) the OWS stage. At the first stage, as the platform has no historical performance information about the vehicles, it has to face the "cold start" problem at the initial selection stage. Inspired by the transfer learning technique, we first investigate the vehicle's daily trajectory penetration and find that its distribution has a weekly periodicity. Based on such periodicity, we can build the temporal feature (i.e., same day in a week) and transfer the knowledge of the vehicles with historical trajectory penetration records to the newly joined vehicles. In this way, the corresponding crowdsourcing performance of the newly joined vehicles can be estimated. In the second stage, we map the CMP to the multiarm bandit problem and adopt the UCB algorithm to solve it.

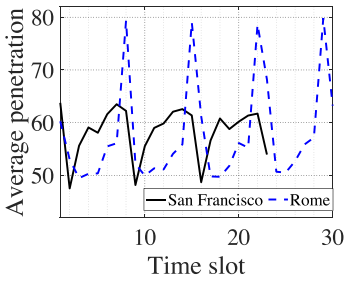

Fig. 7. Average vehicles' trajectory penetration.

A. Transfer Learning-Based Performance Estimation

Given the limited budget, testing all the vehicles is difficult as the platform has to find a way to rapidly accumulate sufficient vehicles' performance data in order to make better decisions for worker selection. In our paper, the performance is directly related to the vehicle daily penetration. As shown in Fig. 7, the average daily trajectory penetration of vehicles in both cities is plotted. It can be seen that the average penetration varies in a similar trend in both cities, indicating that the penetration is periodic in every seven days. We further plot the histogram of the penetration each day. Fig. 8(a) and (b) shows the example in San Francisco and Rome, respectively. By comparing with classic distributions, we discover that the penetration of San Francisco fit the normal distribution

$$f_{\text{no}}(x) = \frac{1}{\sqrt{2\pi}\sigma_t} e^{\left(\frac{-(x-\theta_t)^2}{2{\sigma_t}^2}\right)} \tag{8}$$

with the least error.[6] The penetration of Rome fits the Rayleigh distribution

$$f_{ra}(x) = \frac{x}{{\gamma_t}^2} e^{\left(\frac{-(x)^2}{2{\gamma_t}^2}\right)} \tag{9}$$

with the least error.

Especially, we find that distributions of days under the same temporal context are highly similar, where the temporal context means the day in a week. For instance, the fit distribution and histogram of the 1st and 8th day in San Francisco are shown in Fig. 8(a). We can see that the fitting probability density function (PDF) curves of these two days are very

[6]The fitting process is enabled by the distribution fitter app of MATLAB, and the detailed steps can be found on [37].

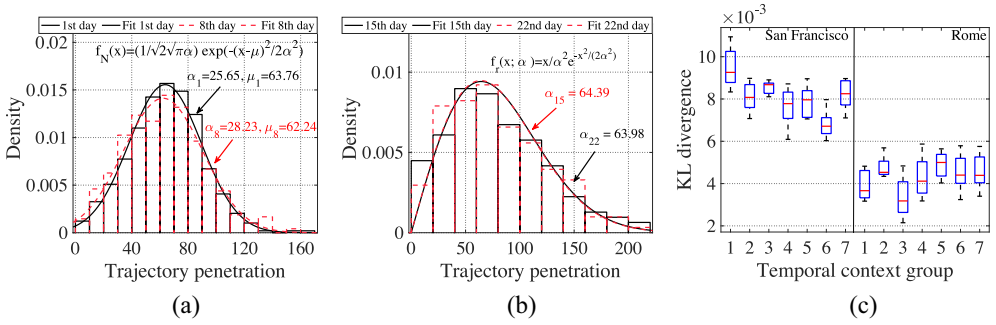

Fig. 8. Vehicle trajectory penetration distribution fit. (a) San Francisco. (b) Rome. (c) KL divergence of fits.

TABLE III
FITTING RESULTS OF VEHICLES' TRAJECTORY PENETRATION IN BOTH CITIES

Fitting Results		Contextual Information Group						
		1	2	3	4	5	6	7
San Francisco	σ^{g*}	62.63±1.20	48.89±1.12	50.77±1.13	58.62±1.09	57.90±1.12	61.28±1.10	62.46±1.10
	θ^g	26.74±0.85	25.66±0.82	25.17±0.80	24.24±0.77	25.05±0.79	24.46±0.78	24.51±0.78
Rome	γ^g	63.82±2.57	46.64±1.76	38.87±1.18	41.02±1.23	42.04±1.25	43.55±1.30	45.31±1.48

* g represents the number of the contextual information group.

close, with the parameter $\{\sigma_1 = 25.65, \theta_1 = 63.76\}$ and $\{\sigma_8 = 28.23, \theta_8 = 62.24\}$ for the 1st and 8th day, respectively. In Rome, the PDF curves of the 15th and 22nd day are almost identical, with the parameter $\{\gamma_{15} = 64.39\}$ and $\{\gamma_{22} = 63.98\}$, respectively. This means that we may infer the penetration of newly joined vehicles by learning the historical vehicle penetration distribution with the same feature (i.e., same temporal context). The fitting results are grouped by the contextual information and shown in Table III. To further verify and quantify such similarity, we calculate the KL divergence between distributions under the same temporal context. For example, we calculate the KL divergence between distributions of the 1st, 8th, 15th, 22nd, and 29th days in Rome as group 1. Due to the weekly periodicity, we have seven groups for each city. The KL divergence can be calculated as

$$D_{\mathrm{KL}}(P||Q) = \int_0^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx \tag{10}$$

where P and Q are distributions of the vehicle trajectory penetration of two days under the same temporal context, where for San Francisco, P and Q follow the normal distribution, and for Rome, they follow the Rayleigh distribution. p and q are the corresponding probability densities of P and Q, respectively. Then, we plot the boxplot of the contextual KL-divergence in Fig. 8(c). We can see that all values of KL divergence are very small. For instance, all values in San Francisco are less than 1.1×10^{-2} with the maximum 1.09×10^{-2} and minimum 6.0×10^{-3}. This confirms that in the same city, distributions under the same temporal context are highly similar.

Therefore, given a limited recruitment budget, we design the *transfer learning-based performance estimation* algorithm as the initial algorithm, which will run a short period while aggregating vehicles' performance (i.e., trajectory penetration) information. Especially, we select T_1 as the length of the initial stage where the value of T_1 is flexible and can be customized according to the platform's need. Moreover, we assume that the incentive cost of each vehicle (i.e., the cost of selecting a worker) is identical [20], [29], and it is set to be one budget unit in our paper. During the first week, we can select K vehicles as the initial vehicles, record the penetration information, and calculate the corresponding performance each time slot. Then, based on these data, we mine the penetration distribution $D_{\bar{t}}$ at each time slot. The corresponding expectation $\kappa_{\bar{t}}$ of each distribution can be calculated, where $\bar{t} = 1, 2, \ldots, 7$ means the temporal context group. For the normal distribution shown in (8), we have

$$\kappa_{\bar{t}}^{\mathrm{no}} = \theta_t. \tag{11}$$

For the Rayleigh distribution in (9), we have

$$\begin{aligned} \kappa_{\bar{t}}^{ra} &= \int_{-\infty}^{+\infty} x f_{ra}(x) dx \\ &= \int_{-\infty}^{+\infty} \frac{x^2}{\gamma_t^2} e^{\left(\frac{-(x)^2}{2\gamma_t^2}\right)} dx \\ &= \sqrt{\frac{\pi}{2}} \gamma_t. \end{aligned} \tag{12}$$

From the 8th to T_1th time slot, we have the worker candidate set at time slot t, $\mathcal{W}_t = \{w_{t,1}, w_{t,2}, \ldots, w_{t,m}, \ldots\}$. $\mathcal{W}_t$ can be divided into the worker set with and without the historical performance: $\mathcal{W}_t^k = \{w_{t,1}^k, w_{t,2}^k, \ldots, w_{t,m_1}^k, \ldots\}$ and $\mathcal{W}_t^{nk} = \{w_{t,1}^{nk}, w_{t,2}^{nk}, \ldots, w_{t,m_2}^{nk}, \ldots\}$, respectively. $\mathcal{W}_t^k \subseteq \mathcal{W}_t$, and $\mathcal{W}_t^{nk} \subseteq \mathcal{W}_t$. The performance of each worker candidate in $\mathcal{W}_t$ is $\bar{\mathcal{P}}_t = \{\pi_{t,1}, \pi_{t,2}, \ldots, \pi_{t,m}, \ldots\}$. Likewise, for $\mathcal{W}_t^k$ and $\mathcal{W}_t^{nk}$, we have $\bar{\mathcal{P}}_t^k = \{\pi_{t,1}^k, \pi_{t,2}^k, \ldots, \pi_{t,m_1}^k, \ldots\}$ and $\bar{\mathcal{P}}_t^{nk} = \{\pi_{t,1}^{nk}, \pi_{t,2}^{nk}, \ldots, \pi_{t,m_2}^{nk}, \ldots\}$, respectively. At time slot t, we first select vehicles with $\pi_{t,m_1}^k > \kappa_{\bar{t}}$, for $w_{t,m_1}^k \in \mathcal{W}_t^k$. Subsequently, we get the first part of the selected worker set $\mathcal{W}_t^1 = \{w_{t,1}^1, w_{t,2}^1, \ldots, w_{t,\bar{m}_1}^1, \ldots\}$, and the corresponding performance set $\bar{\mathcal{P}}_t^1 = \{\pi_{t,1}^1, \pi_{t,2}^1, \ldots, \pi_{t,\bar{m}_1}^1, \ldots\}$.

If $|\mathcal{W}_t^1| < K$, we will then randomly select $K - |\mathcal{W}_t^1|$ vehicles in $\mathcal{W}_t^{nk}$. This part can be defined as $\mathcal{W}_t^2 = \{w_{t,1}^2, w_{t,2}^2, \ldots, w_{t,\bar{m}_2}^2, \ldots\}$, and the corresponding performance

set $\bar{\mathcal{P}}_t^2 = \{\pi_{t,1}^2, \pi_{t,2}^2, \ldots, \pi_{t,\bar{m}_2}^2, \ldots\}$. Because $\mathcal{W}_t^2 \subseteq \mathcal{W}_t^{nk} \subseteq \mathcal{W}_t$ and $\mathbb{E}[\pi_{t,m}] \sim \mathcal{D}_{\bar{\imath}}$, we can assume that $\mathbb{E}[\pi_{t,\bar{m}_2}^2] \sim \mathcal{D}_{\bar{\imath}}$. All the selected vehicles' performance will be recorded, and $\mathcal{W}_t^k$ and $\mathcal{W}_t^{nk}$ will be continuously updated along the algorithm processing. The above loop is executed until the T_1th time slot. Such operations enable the platform to maintain a high-level collection during the initial stage, thereby successfully solving the "cold start" problem in VBC.

Theorem 1: The transfer learning-based performance estimation algorithm can achieve a nondecreasing collection utility.

Proof: For $\forall w_{t,\bar{m}_1}^1 \in \mathcal{W}_t^1$, $\pi_{t,\bar{m}_1}^1 > \kappa_{\bar{\imath}}$, we have

$$\frac{1}{\left|\mathcal{W}_t^1\right|} \sum_{w_{t,\bar{m}_1}^1 \in \mathcal{W}_t^1} \pi_{t,\bar{m}_1}^1 = \kappa_2 \geq \kappa_{\bar{\imath}} \tag{13}$$

where $|\mathcal{W}_t^1| \leq K$. If $|\mathcal{W}_t^1| = K$, then the theorem is proved. If $|\mathcal{W}_t^1| < K$, because $\mathbb{E}[\pi_{t,\bar{m}_2}^2] \sim \mathcal{D}_{\bar{\imath}}$, we have

$$\left|\mathcal{W}_t^1\right| + \left|\mathcal{W}_t^2\right| = K \tag{14}$$

$$\mathbb{E}\left[\frac{1}{\left|\mathcal{W}_t^2\right|} \sum_{w_{t,\bar{m}_2}^2 \in \mathcal{W}_t^2} \pi_{t,\bar{m}_2}^2\right] = \kappa_1 = \kappa_{\bar{\imath}}. \tag{15}$$

By combining (13)–(15), we have

$$\frac{\kappa_1 * \left|\mathcal{W}_t^2\right| + \kappa_2 * \left|\mathcal{W}_t^1\right|}{K} > \kappa_{\bar{\imath}} \tag{16}$$

which concludes the proof. ∎

Therefore, the transfer learning-based performance estimation method can help the platform to achieve a nondecreasing crowdsourcing performance in the initial stage.

B. OWS Algorithm

Based on the initialization, from the $(T_1 + 1)$th time slot to the end, we can characterize the worker selection problem as an E2 process. Then, we propose the OWS to solve the problem by adopting the UCB algorithm. Finally, we give a straightforward guideline for adjusting the level of exploration of the OWS.

1) Problem Mapping: We map the worker selection problem as a classic multiarm bandit process. At each time slot, the platform makes the selection decision (i.e., a certain arm is pulled or not) and observes every worker selection (i.e., slot machine) to obtain the corresponding worker utility (i.e., reward). For the collection during the long term, we aim at maximizing the cumulative platform utility (i.e., cumulative reward). To this end, E2 can be an efficient approach to characterize the problem, which jointly considers the decision making (exploitation) and the quality measuring (exploration). Based on E2, we can dynamically exploit workers who return the result with a higher quality and explore workers whose performance has high uncertainty.

2) UCB-Based Worker Selection: In particular, based on the UCB algorithm [38], we design a K-greedy approach to solve our problem. In the approach, the workers who can achieve the top K highest UCB of the worker utility will be opportunistically selected. For each time slot t, the worker utility of worker $w_{t,m}$ can be calculated by (2). Then, the average worker utility of worker $w_{t,m}$ till time slot t is

$$\bar{\pi}_{t,m} = \frac{1}{t} \sum_{t \in \mathcal{T}} \pi_{t,m} \tag{17}$$

and the upper confidence around the mean value $\bar{\pi}_{t,m}$ can be calculated as

$$b_{t,m} = \mu \sqrt{\frac{\ln t}{f_{t,m}}} \tag{18}$$

where $f_{t,m}$ is the selecting frequency of worker $w_{t,m}$ till time slot t, and μ is the exploration factor that balances the E2 process.

With the above setting, there is a high probability that at the next time slot $t+1$, the true performance value $\pi_{t+1,m}$ of $w_{t,m}$ lies in the confidence interval $[\bar{\pi}_{t,m} - b_{t,m}, \bar{\pi}_{t,m} + b_{t,m}]$. This can be obtained according to the Chernoff–Hoeffding inequality as follows.

Lemma 1 (Chernoff–Hoeffding Inequality): Let $Z_1, \ldots, Z_n$ be bounded random variables with $Z_i \in [a, b]$ for all i, where $-\infty < a < b < +\infty$ and $b - a = \Delta$. $\mathbb{E}[Z_t | Z_1, Z_2, \ldots, Z_{t-1}] = \omega$. Suppose that $\Omega_n = \sum_{i=1}^n Z_i$. Then, for $\forall d > 0$, we will have $\mathbb{P}\{\Omega_n \geq n\omega + d\} \leq e^{[(-2d^2 n)/\Delta^2]}$ and $\mathbb{P}\{\Omega_n \leq n\omega - d\} \leq e^{[(-2d^2 n)/\Delta^2]}$.

From the above lemma, we can get

$$\mathbb{P}\left\{\left|\pi_{t+1,m} - \bar{\pi}_{t,m}\right| \geq b_{t,m}\right\} \leq 2t^{\frac{-2\mu^2}{\Delta^2}} \tag{19}$$

Algorithm 1 OWS Algorithm

Input: : worker candidate set $\mathcal{W}_t = \{w_{t,1}, w_{t,2}, \ldots, w_{t,m}, \ldots\}$, corresponding emperical performance $\bar{\mathcal{P}}_t = \{\pi_{t,1}, \pi_{t,2}, \ldots, \pi_{t,m}, \ldots\}$, selecting frequency $f_{t,m}$, and Budget K. The termination time is set to be T_2.

Output: : Selected worker set $\mathcal{W}_t^s = \{w_{t,1}^s, w_{t,2}^s, \ldots, w_{t,m}^s\}$.

After initialization: $\mathcal{W}_t^s \leftarrow \phi$, $t \leftarrow T_1 + 1$, $i \leftarrow 1$
for $t < T_2$ **do**
  update $b_{t,m} = \mu \sqrt{\frac{\ln t}{f_{t,m}}}$
  $\pi_{t+1,m} = \overline{\pi}_{t,m} + b_{t,m}$
  $t = t + 1$
  while $i < K + 1$ **do**
    select $w_{t,i}^s = \arg \max_{w_{t,m} \in \{W_t \setminus \mathcal{W}_t^s\}} \pi_{t,m}$
    $\mathcal{W}_t^s = \mathcal{W}_t^s \cup \{w_{t,i}^s\}$
    $i = i + 1$
  end while
  for $w_{t,m} \in \mathcal{W}_t^s$ **do**
    $f_{t,m} = f_{t,m} + 1$
    observe instant performance $\hat{\pi}_{t,m}$
    update $\overline{\pi}_{t,m} = \frac{f_{t-1,m}\overline{\pi}_{t-1,m} + \hat{\pi}_{t,m}}{f_{t,m}}$
  end for
  for $w_{t,m} \notin \mathcal{W}_t^s$ **do**
    $f_{t,m} = f_{t-1,m}$, $\overline{\pi}_{t,m} = \overline{\pi}_{t-1,m}$
  end for
end for

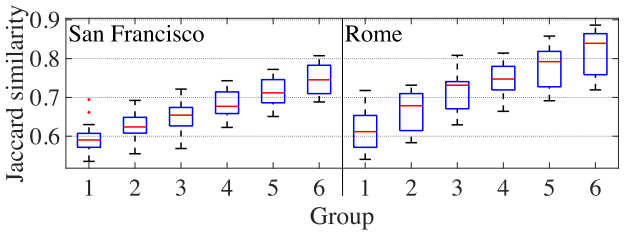

Fig. 9. Jaccard similarity of vehicles in both cities.

where with $\mu > 1$, the probability of $\pi_{t+1,m}$ outside the confidence interval tends to infinitesimal rapidly.

As shown in Algorithm 1, at each time slot, we will update the upper confidence value $b_{t,m}$ and the expected reward $\bar{\pi}_{t,m}$ for each worker (lines 1–4). Then, we will calculate the upper bound performance $\bar{\pi}_{t,m} + b_{t,m}$ and accordingly choose the workers who can achieve top K performance (i.e., upper bound confidence value, lines 5–18). Equations (17) and (18) indicate that two conditions can lead to the increase of a worker's upper bound performance: 1) the worker rarely been selected by the platform, the upper confidence value $b_{t,m}$ will increase quickly as the time slot t going and 2) the worker achieves a good collection performance, i.e., the high value of $\bar{\pi}_{t,m}$. Therefore, given a crowdsourcing task, our scheme can select the current good-performance workers, meanwhile be optimistic about choosing the workers with high uncertainty, as the corresponding performance has a high probability locating in the proved confidence interval.

C. Guidance of Selecting Exploration Parameter μ

In the OWS algorithm, the exploration parameter μ is critical, which determines how much exploration will be taken during the worker selection process. Particularly, we give insights of the workers' performance to guide platforms to choose the suitable μ. We can generally measure a worker's performance (i.e., trajectory penetration) from two aspects: 1) collection quantity: $v_{t,m}$, the collection quantity of worker $w_{t,m}$ at time slot t and 2) collection coverage (measured by entropy): H_t^m, the entropy of worker $w_{t,m}$ at time slot t. The worker with high trajectory penetration (i.e., high collection quantity and wide coverage) will be selected with high probability. Furthermore, we define $\mathcal{J}_{r,\text{tra}}^t$ and $\mathcal{J}_{r,\text{cov}}^t$ as the set of the top r workers in terms of the collection quantity and coverage, respectively, at time slot t. Then, the Jaccard similarity coefficient between $\mathcal{J}_{r,\text{tra}}^t$ and $\mathcal{J}_{r,\text{cov}}^t$ can be calculated as

$$J_r^t = \frac{|\mathcal{J}_{r,\text{tra}}^t \cap \mathcal{J}_{r,\text{cov}}^t|}{|\mathcal{J}_{r,\text{tra}}^t \cup \mathcal{J}_{r,\text{cov}}^t|}. \tag{20}$$

Intuitively, a higher Jaccard similarity value indicates more overlap between the set $\mathcal{J}_{r,\text{tra}}^t$ and $\mathcal{J}_{r,\text{cov}}^t$. We present the box plot of Jaccard similarity of San Francisco and Rome in Fig. 9. The parameter r ranges from 60 to 160 (i.e., top 60 to top 160) with a step length 20, totally six groups. It can be found that the Jaccard similarity value increases as the value of r increases. For instance, as r grows, the median value of similarity increases from 0.59 to 0.75 and 0.61 to 0.84 in San Francisco and Rome, respectively. This demonstrates that when r is large, the corresponding top-r set of collection quantity and coverage is consistent. Therefore, with the larger budget, when the platform selects a worker with a high collection quantity, the worker is more likely to perform well in terms of the coverage as well. In this case, the platform can select a lower exploration parameter during the OWS process μ, and *vice versa*. The above instruction can be readily applied to different scenarios (e.g., different target areas and different budgets), and help platforms to configure the level of exploration.

VI. Performance Evaluation

In this section, we evaluate the proposed *POSE* by extensive trace-driven simulations. Particularly, we first present the experiment setup and design the benchmark strategies. Then, we give the performance metrics and carry out the evaluation performance of *POSE* during both the initial stage and selections stage. Finally, we investigate the impact of the budget and exploration factor μ.

A. Methodology

Experimental Setup: For the long-term performance evaluation, we first, respectively, choose three-weeks data and four-week data in San Francisco and Rome for the data extension and accordingly get two 2520-days (i.e., 360-weeks)[7] extended traces for both two cities. Our *POSE* scheme has two stages. The first stage is the initialization stage, which starts from time slot 1 and ends at time slot T_1. The second stage is the selection stage, which starts from time slot $T_1 + 1$ and ends at time slot T_2. $T_1 = 30$ and $T_2 = 2520$ in the evaluation. To investigate the impact of the budget, we fix the exploration factor at $\mu = 10$ and vary the budget from 20 to 50 with a step size of 10. To reveal the impact of the exploration factor μ, we vary the budget K to 20, 30, 40, and 50, and the exploration factor μ to 1, 10, 50, and 100 [39].

Benchmark Schemes: We design four benchmark schemes to compare with our *POSE* scheme as follows.

1) *Random:* Regardless of their historical performance, this scheme randomly selects workers with the same probability.
2) *Dynamic-Greedy (Greedy):* In this scheme, the average worker utility of each vehicle will be updated dynamically. Then, the scheme will adaptively select the vehicles who have the top-K highest worker utilities to be the workers for the next time slot. It is worth noting that the average worker utility will be updated only when the corresponding worker is selected.
3) *UCB-Greedy (UCBG):* This scheme is the closest scheme to ours [19]. In this scheme, the UCB algorithm works in a greedy way without the initial stage. UCBG greedily selects the workers with the top-K highest upper confidence value at each time slot.
4) *Oracle:* In this scheme, we assume that the future long-term vehicles' trajectory penetration is known ahead, and

[7]Due to the lack of data, we derive the long-term data by data extension. The length of the extended data can be flexibly adjusted, which does not have an effect on the overall performance.

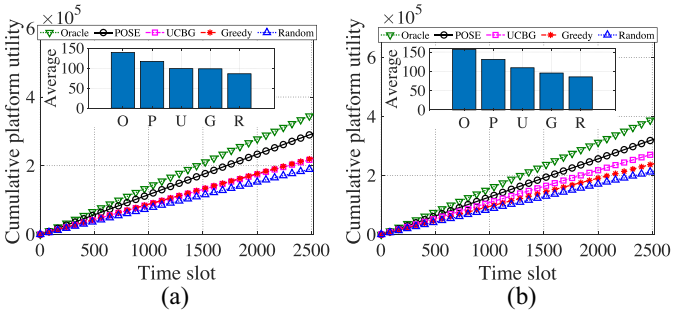

Fig. 10. Overall performance. (a) San Francisco. (b) Rome.

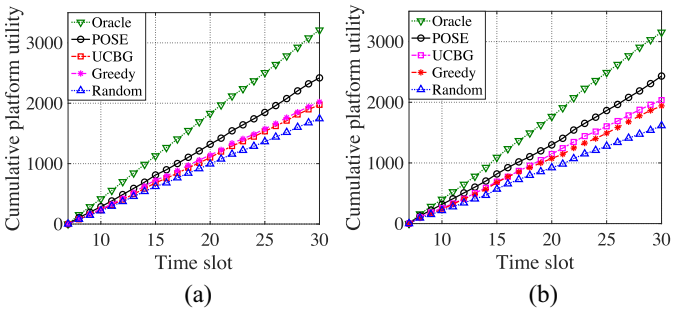

Fig. 11. Performance of the initial stage. (a) San Francisco. (b) Rome.

the worker selection can be decided accordingly. In reality, this scheme cannot be achieved and we adopt it as an upper bound for the performance comparison.

Performance Metrics: We define the platform utility as the metric to evaluate the worker selection performance in VBC. The platform utility refers to the cumulative utility of selected workers to the platform. It is calculated by the sum of the worker utility of the selected workers each day.

B. Performance Comparison

We first set the budget K and the exploration factor μ to 30 and 10, respectively.

1) High-Performance Worker Selection: Fig. 10 shows the results of the cumulative platform utility over time slots, and we can observe that the proposed *POSE* outperforms the UCBG, greedy, and random in both two cities, and has a gap from the Oracle. Specially, we plot the average platform utility each day, and it can be found that compared with the average platform utility of 98.8 and 109.0 in UCBG, 98.4 and 95.6 in greedy, and 86.0 and 85.6 in random, the *POSE* can, respectively, achieve 117.0 and 131.1 in San Francisco and Rome. Such a performance can improve the collecting performance by 18.4% and 20.3% for UCBG, 18.9% and 37.1% for greedy, and 36.0% and 53.2% for random in San Francisco and Rome, respectively. The reason for the superior performance can be attributed to the mechanism of *POSE*. First, compared with the simple exploitation approach in greedy, *POSE* has an online learning process with both the exploitation and exploration during the selection, and thus it can strike a better performance when facing the worker candidates with dynamic and unstable behaviors across the time span. Second, compared with the UCBG, the transfer learning-based performance estimation

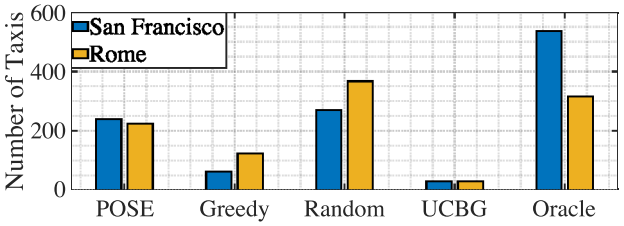

Fig. 12. Number of known vehicles after initial stage.

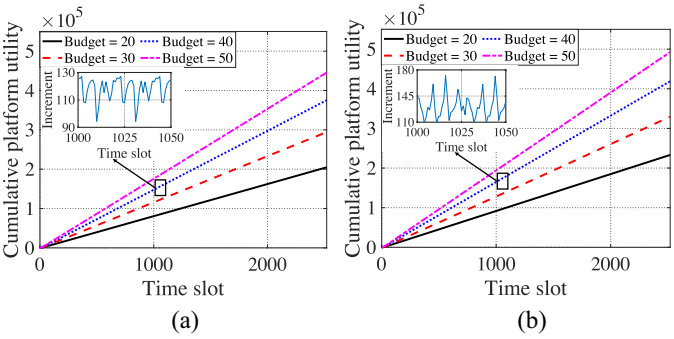

Fig. 13. Impact of different budgets. (a) San Francisco. (b) Rome.

makes *POSE* have a better data aggregation than UCBG and thus a larger exploitation and exploration space. However, the future vehicles' performance in the urban area is highly uncertain, especially for taxis, therefore there is a utility gap from the Oracle.

2) Effective Initializaion: We then check the performance of the initial stage and plot the corresponding cumulative utility of different schemes in Fig. 11. From the figure, we can see that in San Francisco, *POSE* achieves the cumulative platform utility of 2420, and respective improves 22.6%, 20.2%, and 38.8% compared with 1974 of UCBG, 2023 of greedy, and 1744 of random. In Rome, *POSE* achieves the cumulative platform utility of 2432, and respective improves 19.4%, 25.3%, and 50.5% compared with 2036 of UCBG, 1940 of greedy, and 1615 of random. However, we can also find that the performance gap between the *POSE* and Oracle in the initial stage is larger than the overall period. This is because in the initial stage, *POSE* does not have enough vehicles' trajectory penetration information to use, resulting in unsatisfied learning results and performance. To figure out the initializing ability, we further plot the number of known vehicles aggregated after the initial stage. As shown in Fig. 12, *POSE* respective aggregated 240 and 224 vehicles' information, in San Francisco and Rome. Compared with the 63 and 124 of greedy, 367 and 270 of random, 30 and 30 of UCBG, and 536 and 316 of Oracle, *POSE* respective achieves the 2.8× and 0.8×, 7×, and 6.5× improvement. Moreover, the number of known vehicles can be further improved by extending the time span of the initial stage.

C. Impact of Budget

The recruitment budget is an important factor of crowdsourcing. Fig. 13 presents the performance of *POSE* with varying budget amounts, where the exploration factor is set

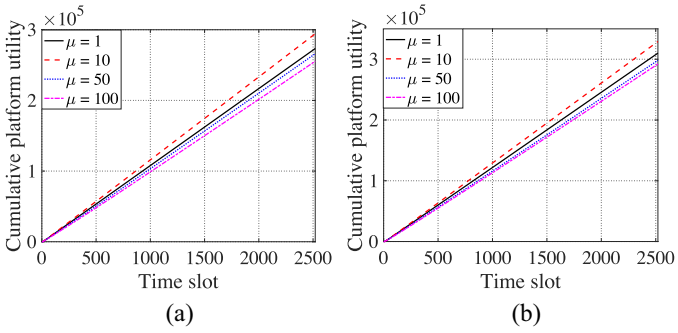

Fig. 14. Impact of exploration μ. (a) San Francisco. (b) Rome.

to be 10. We can find that the budget amount has a positive correlation with the cumulative platform utility, where the larger budget amount can perform a higher cumulative platform utility. While, it is interesting to observe that the utility improvements between different budgets decrease when the budget amount increases. At the end of crowdsourcing, in San Francisco and Rome, it is 43.2% and 41.7% improvement between budget 30 and budget 20, 27.5% and 27.2% improvement between budget 40 and budget 30, and 18.7% and 17.8% improvement between budget 50 and budget 40, respectively. Such a phenomenon is possibly caused by the submodular effect, where with the map collection going, the new vehicle has a high probability of covering the same road that has been collected before. Therefore, the redundancy degree will increase and slow utility improvement. Moreover, small plots in the figure indicate that the increment of the cumulative platform utility slightly fluctuates over the period, and the plot of the cumulative platform utility is not linear.

D. Impact of Exploration μ Factor

The exploration factor μ determines how much exploration will be taken in the worker selecting process. Fig. 14 shows the algorithm's performance under varying values of μ. The budget is set to be 30, and μ is set to be 1, 10, 50, and 100. In Fig. 14(a) and (b), we can find that the best algorithm's performance is achieved at μ of 10. From this finding, we can infer that the suitable value of μ needs to be well optimized, and a larger value will not necessarily result in a better performance. A suitable value of μ can perform a desirable tradeoff and enable a balanced exploration, where the algorithm can find the potential high-performance workers. However, if the selection scheme takes excessive or insufficient exploration, the corresponding collection performance will be also significantly lowered. As indicated in Section V-C, platforms can adjust the value of μ according to the Jaccard similarity.

E. Reducing the Gap Between POSE and the Oracle

During the long-term collection, possible dynamics caused by weather, special social events, etc., may slightly change the distribution of vehicles' daily trajectory penetration. This can degrade the performance of transfer learning-based estimation. A sliding window mechanism may solve the problem, and reduce the utility gap between the *POSE* and the Oracle. If we adopt a sliding window with a size of L and the contextual distribution within the sliding window can be considered as stable. As the selection process evolves, we gradually move the sliding window forward so that we can get a more accurate contextual distribution for the next selection round.

VII. Conclusion

In this article, we have investigated the edge-enabled VBC toward HD map collection for AVs. Specifically, we have formulated the CMP to maximize the cumulative utility. To solve the problem, we have characterized the trajectory penetration of vehicles in terms of distribution, correlation, and variation, which collectively lead to our *POSE* design. In *POSE*, we have integrated two major components: 1) transfer learning-based performance estimation and 2) the OWS. Extensive trace-driven simulations have been conducted and the results have demonstrated the efficiency of *POSE* in map collection. We believe the systematic principle of *POSE* can also be readily applied to other VBC scenarios, e.g., traffic monitoring, road anomaly detection, etc. For our future work, in addition to the HD map collection, we will also investigate the HD map caching strategy and delivery optimization.

References

[1] X. Cao *et al.*, "Online worker selection towards high quality map collection for autonomous driving," in *Proc. IEEE Globecom*, Dec. 2019, pp. 1–6.
[2] *Rand Autonomous Vehicles*. Accessed: Apr. 2019. [Online]. Available: https://www.rand.org/topics/autonomous-vehicles.html
[3] T. Jiang, H. Fang, and H. Wang, "Blockchain-based Internet of Vehicles: Distributed network architecture and performance analysis," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4640–4649, Oct. 2018.
[4] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.
[5] L. Chen, X. Hu, W. Tian, H. Wang, D. Cao, and F.-Y. Wang, "Parallel planning: A new motion planning framework for autonomous driving," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 1, pp. 236–246, Jul. 2018.
[6] Q. Cui *et al.*, "Big data analytics and network calculus enabling intelligent management of autonomous vehicles in a smart city," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2021–2034, Sep. 2018.
[7] Y. Zheng, J. Wang, and K. Li, "Smoothing traffic flow via control of autonomous vehicles," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3882–3896, Jan. 2020.
[8] F. Lyu *et al.*, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020.
[9] F. Bock, S. Di Martino, and A. Origlia, "Smart parking: Using a crowd of taxis to sense on-street parking space availability," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 496–508, Mar. 2019.
[10] X. Zhang, Z. Yang, and Y. Liu, "Vehicle-based bi-objective crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3420–3428, Jan. 2018.
[11] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 915–929, Apr. 2019.
[12] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: A survey," *VLDB J.*, vol. 29, no. 1, pp. 1–34, Aug. 2019.
[13] X. Wang, W. Wu, and D. Qi, "Mobility-aware participant recruitment for vehicle-based mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4415–4426, May 2018.
[14] K. Yi, R. Du, L. Liu, Q. Chen, and K. Gao, "Fast participant recruitment algorithm for large-scale vehicle-based mobile crowd sensing," *Pervasive Mobile Comput.*, vol. 38, pp. 188–199, Jul. 2017.
[15] S. Sarker, M. A. Razzaque, M. M. Hassan, A. Almogren, G. Fortino, and M. Zhou, "Optimal selection of crowdsourcing workers balancing their utilities and platform profit," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8602–8614, Jun. 2019.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[17] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. L. Re, "IncentMe: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1571–1584, Aug. 2018.

[18] N. Cheng *et al.*, "Big data driven vehicular networks," *IEEE Netw.*, vol. 32, no. 6, pp. 160–167, Aug. 2018.

[19] P. Yang, N. Zhang, S. Zhang, K. Yang, L. Yu, and X. Shen, "Identifying the most valuable workers in fog-assisted spatial crowdsourcing," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1193–1203, Oct. 2017.

[20] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *J. Mach. Learn. Res.*, vol. 3, pp. 397–422, Nov. 2002.

[21] X. Wang, R. Jia, X. Tian, X. Gan, L. Fu, and X. Wang, "Location-aware crowdsensing: Dynamic task assignment and truth inference," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 362–375, Nov. 2018.

[22] X. Gao, S. Chen, and G. Chen, "MAB-based reinforced worker selection framework for budgeted spatial crowdsensing," *IEEE Trans. Knowl. Data Eng.*, early access, May 4, 2020, doi: 10.1109/TKDE.2020.2992531.

[23] W. Ma and S. Qian, "High-resolution traffic sensing with autonomous vehicles," Oct. 2019. [Online]. Available: arXiv:1910.02376.

[24] F. Morselli, F. Zabini, and A. Conti, "Environmental monitoring via vehicular crowdsensing," in *Proc. IEEE PIMRC*, Sep. 2018, pp. 1382–1387.

[25] C.-W. Yi, Y.-T. Chuang, and C.-S. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1905–1917, Mar. 2015.

[26] L. Tang, X. Yang, Z. Dong, and Q. Li, "CLRIC: Collecting lane-based road information via crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2552–2562, Mar. 2016.

[27] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Reputation-aware, trajectory-based recruitment of smart vehicles for public sensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1387–1400, Aug. 2017.

[28] M. Hu, Z. Zhong, Y. Niu, and M. Ni, "Duration-variable participant recruitment for urban crowdsourcing with indeterministic trajectories," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10271–10282, Jun. 2017.

[29] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 2542–2550.

[30] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2982–2997, Apr. 2018.

[31] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, "iLocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1831–1847, Aug. 2020.

[32] J. Huang *et al.*, "Blockchain based mobile crowd sensing in industrial systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6553–6563, Oct. 2000.

[33] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi. (2014). *CRAWDAD Dataset ROMA/Taxi (V.2014-07-17)*. [Online]. Available: https://crawdad.org/roma/taxi/20140717

[34] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. (2009). *CRAWDAD Dataset EPFL/Mobility (V.2009-02-24)*. [Online]. Available: https://crawdad.org/epfl/mobility/20090224

[35] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *Amer. Stat.*, vol. 42, no. 1, pp. 59–66, Feb. 1988.

[36] F. Lyu *et al.*, "LEAD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 6553–6563, Oct. 2020, doi: 10.1109/TMC.2020.2984261.

[37] *MATLAB Distribution Fitter App*. Accessed: Apr. 2019. [Online]. Available: https://www.mathworks.com/help/stats/model-data-using-the-distribution-fitting-tool.html

[38] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *Proc. Int. Conf. Algorithm. Learn. Theory*, Oct. 2011, pp. 174–188.

[39] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Jul. 2018.

Xiaofeng Cao (Graduate Student Member, IEEE) received the bachelor's and master's degrees from the National University of Defense Technology, Changsha, China, in 2014 and 2016, respectively, where he is currently pursing the Ph.D. degree with the College of Systems Engineering.

He was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from September 2018 to September 2020. His research mainly focuses on cloud/edge computing and data-driven application design.

Mr. Cao is a member of the IEEE Computer Society, Communication Society, and Vehicular Technology Society.

Peng Yang (Member, IEEE) received the B.E. degree in communication engineering and the Ph.D. degree in information and communication engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013 and 2018, respectively.

He was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from September 2015 to September 2017, and a Postdoctoral Fellow from September 2018 to December 2019. Since January 2020, he has been a Faculty Member with the School of Electronic Information and Communications, HUST. His current research focuses on mobile-edge computing, video streaming, and analytics.

Feng Lyu (Member, IEEE) received the B.S. degree in software engineering from Central South University, Changsha, China, in 2013, and the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2018.

From September 2018 to December 2019 and from October 2016 to October 2017, he worked as a Postdoctoral Fellow and was a visiting Ph.D. student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently a Professor with the School of Computer Science and Engineering, Central South University. His research interests include vehicular networks, beyond 5G networks, big data measurement and application design, and cloud/edge computing.

Prof. Lyu is a recipient of the Best Paper Award of IEEE ICC 2019. He currently serves as an Associate Editor for IEEE SYSTEMS JOURNAL and a Leading Guest Editor for *Peer-to-Peer Networking and Applications*, and served as TPC member for many international conferences. He is a member of the IEEE Computer Society, IEEE Communication Society, and IEEE Vehicular Technology Society.

Jiarong Han received the bachelor's degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2014, and the master's degree from the National University of Defense Technology, Changsha, China, in 2016.

He is currently a Research Assistant with Beijing University of Posts and Telecommunications. His research mainly focuses on data analytics and machine learning.

Yan Li received the bachelor's degree from the National University of Defense Technology, Changsha, China, in 2014, where she is currently pursing the Ph.D. degree with the College of Systems Engineering.

Her research interests include resource management and network traffic analysis in data center networking.

Deke Guo (Senior Member, IEEE) received the B.S. degree in industry engineering from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2008.

He is currently a Professor with the College of System Engineering, National University of Defense Technology, and is also with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks.

Prof. Guo is a member of ACM.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Dr. Shen received the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the elected IEEE Communications Society Vice President for Technical & Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a Member of IEEE ComSoc Fellow Selection Committee. He was/is the Editor-in-Chief of IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, *IET Communications*, and *Peer-to-Peer Networking and Applications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.