IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 July 2023, accepted 19 August 2023, date of publication 24 August 2023, date of current version 31 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3308202

RESEARCH ARTICLE

Context-Aware Worker Recruitment for Mobile Crowd Sensing Based on Mobility Prediction

QUAN T. NGO AND SEOKHOON YOON, (Member, IEEE)
Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea

Corresponding author: Seokhoon Yoon (seokhoonyoon@ulsan.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2021R1I1A3051364, and in part by the Ministry of Science and ICT under Grant 2019R1F1A1058147.

ABSTRACT Opportunistic worker (OW) selection is a challenging problem in mobile crowd sensing (MCS), where tasks are assigned to individuals to be completed seamlessly during their daily routines without any deviation from their usual routes. In this paper, we propose a novel framework named context-aware worker recruitment based on a mobility prediction model (CAMP) to address the OW selection problem in MCS. Unlike previous approaches that relied on worker mobility prediction models with limited accuracy or utility-based selection methods neglecting task distribution differences across locations, CAMP introduces a two-phase strategy for OW selection. In the first phase, we leverage a recurrent neural network-based prediction model specifically designed to forecast volunteer workers' future locations with higher precision. This enhanced mobility prediction ensures more effective task assignments in the MCS system. In the second phase, CAMP employs a weighted-utility algorithm that takes into account the varying task distribution throughout the day across different locations. The key novelty of the CAMP framework lies in its combination of an accurate multi-output RNN model for predicting worker mobility and a unique weighted-utility worker selection algorithm that considers variations in task distribution across different locations and sensing cycles. To validate the effectiveness of the CAMP framework, we extensively evaluate it using real-world GPS data, specifically the Crawdad Roma/Taxi dataset. The results demonstrate that CAMP outperforms existing approaches, delivering a higher number of completed tasks while adhering to the same budget constraints.

INDEX TERMS Human mobility prediction, opportunistic worker selection, mobile crowd sensing, task allocation.

I. INTRODUCTION

Mobile crowd sensing (MCS) is a promising method for gathering urban data that involves dynamically moving mobile users (known as workers). In MCS, workers use their mobile devices to provide environmental, economic, and social data. MCS has gained the interest of both academia and industry due to its potential to provide accurate, cost-effective, and scalable sensing solutions [1], [2]. As a result, MCS has encouraged various environmental, economic, and social applications [3], [4].

Task allocation in MCS involves strategically assigning sensing tasks to mobile workers to efficiently collect data in urban areas [5]. It optimizes worker selection based on mobility, expertise, and availability for timely task completion. This enhances MCS project performance, impacting data quality, completion rates, and resource use. Various approaches consider factors like worker mobility prediction, task traits, and geography. Workers are categorized as participatory (move to task locations) or opportunistic (integrate tasks into routines). Participatory workers ensure task completion but face limitations [6], [7]. Workers must deviate from their regular schedules and travel to specific locations to complete the tasks, which can be inconvenient and costly. Even if they are willing to contribute sensing data, the cost may deter some workers from participating. Furthermore, the cost of MCS projects in participatory mode is frequently higher from providing incentives to cover workers' travel expenses.

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

Opportunistic workers (OWs) play an important role in MCS [8], [9]. The OWs offer an efficient and cost-effective approach, as workers contribute data seamlessly while going about their routine schedules, requiring no additional travel or effort. This mode of involvement maximizes resource utilization and potentially increases the available pool of workers for MCS projects. However, OWs selection can be difficult because the accuracy and effectiveness of collected data can be highly dependent on customary routes and tasks in frequently visited areas. To assign tasks effectively to OWs, it is frequently necessary to predict workers' routes, which can be challenging due to the complexity and unpredictability of real-life factors. As a result, worker selection based on opportunistic mode may be less accurate than intended, and data quality for some tasks may be lower than expected.

This paper proposes a new framework called context-aware worker recruitment based on mobility prediction (CAMP) for selecting OWs for urban environment MCS sensing projects. CAMP employs a two-phase strategy, the first of which involves predicting future locations based on workers' historical mobility data. The second phase assigns tasks to workers using a weighted-utility function that accounts for dynamic changes in task distribution throughout the day.

In the first phase, we introduce a more advanced approach by employing a recurrent neural network (RNN)-based model for worker mobility prediction. RNNs have proven to be highly effective in handling time series data and capturing the intricate patterns of worker movement [10], [11], [12]. By leveraging the power of RNNs, CAMP overcomes the shortcomings of the inhomogeneous Poisson process assumption, enabling the framework to make predictions more accurately and select workers with a higher probability of visiting locations with a substantial number of tasks. This incorporation of an RNN-based prediction model contributes to the overall effectiveness of CAMP in selecting opportune workers, further improving the performance of MCS tasks.

In the second phase, we present a novel weighted-utility worker selection algorithm designed to address the OWs selection problem. While previous methods like Icrowd [13] ignores variations in the number of tasks at each location or DLMV [14] assumes a constant number of tasks at all locations over time, our proposed algorithm takes dynamic changes in task distribution into account. Specifically, it prioritizes locations projected to have more tasks in the next sensing cycle (SC) and selects workers with a higher likelihood of visiting these locations. This intelligent approach enables the algorithm to efficiently identify workers best suited for completing a higher number of tasks, enhancing the overall performance of the MCS project.

The proposed framework was evaluated using a GPS-based dataset called the Crawdad Roma/Taxi dataset [15]. The results show that the proposed framework can outperform baseline models in the number of completed tasks for the same budget amount.

Our contributions are summarized as follows:

- The CAMP framework is proposed for the selection of OWs in MCS projects.
- The RNN-based multi-output human mobility prediction model accurately predicts worker locations in the next sensing cycle.
- The weighted-utility worker selection algorithm recruits a set of OWs who can complete the maximum number of sensing tasks under budget constraints.
- The proposed framework is evaluated with a GPS-based taxi dataset, and the results demonstrate that CAMP outperforms existing methods for OWs selection in MCS, providing an effective solution for this task.

The remainder of this article is organized as follows. Section II introduces related work. Section III details the system model and problem definition. Section IV describes our CAMP framework. Section V presents experiment results from CAMP and other baseline methods. Section VI gives concluding remarks.

II. RELATED WORKS

A. WORKER SELECTION

Worker selection is an important process in MCS that can have a significant impact on the quality and reliability of the data collected. MCS systems typically rely on a large number of workers, many of whom are volunteers, to complete tasks like collecting data and conducting surveys. It is critical to select the best workers for these jobs in order to ensure that the data collected are accurate and useful. For worker selection in MCS, a variety of approaches have been proposed, including heuristics based on worker attributes such as reliability and expertise, as well as machine learning-based approaches that can take into account a wide range of factors. Overall, the goal of MCS worker selection is to find the best match between workers and tasks in order to maximize the quality and utility of the data collected.

One area of study focuses on using a participatory worker approach in which MCS servers require workers to change their routes and go to specific locations to complete sensing tasks. This approach employs two task assignment models: worker-selected tasks (WST), in which workers select their own tasks, and server-assigned tasks (SAT), in which tasks are automatically assigned by the MCS server [16]. The WST model gives workers more freedom, allowing them to choose tasks that match their interests and goals, which can boost motivation and involvement, resulting in higher-quality data collection [17], [18], [19]. However, the WST model may result in reduced coverage and inefficiency in task completion, because workers might not select tasks that are critical to the MCS system's overall performance.

The SAT model, on the other hand, centralizes the task selection and assignment process, allowing the MCS server to optimize task selection based on factors like coverage, quality, and cost [20], [21], [22]. This can lead to more efficient task completion and improved overall MCS system performance. However, because workers have no say in the tasks assigned to them, the SAT-based approach may result in decreased worker motivation and engagement. In order to fairly distribute tasks and optimize task completion, the SAT model also necessitates a complex algorithm for task allocation and management. As a result, researchers have been looking into ways to strike a balance between worker autonomy and MCS system performance using various strategies such as task incentives, worker preferences, and adaptive task assignments.

Another area of study is the use of OWs, who can provide sensing data as part of their daily routines without deviating from their original paths [23], [24], [25]. Most of this research concentrates on selecting OWs for a single data-sensing task with a specific goal and budget constraints. For example, researchers investigated worker recruitment for a single data sensing task and presented various worker selection strategies to select a minimum set of workers to ensure a certain level of sensing quality or to select a specific number of workers to optimize data quality. Other studies have focused on maximizing the overall benefits of multiple concurrent data collection tasks on an MCS platform designed for multiple tasks that share limited resources. These studies propose algorithms for task allocation that optimize overall system performance when tasks are constrained by a limited incentive budget or when multiple tasks must share a pool of workers with limited data collection capabilities.

To address the challenges associated with OW selection and enhance data collection efficiency, the incorporation of mobility prediction models becomes crucial [26], [27]. Mobility prediction involves estimating how OWs will move in the future based on their past locations or relevant data. By accurately predicting their routes, MCS systems can optimize task assignment to OWs, thereby improving data collection efficiency and accuracy. However, predicting worker mobility in real-life situations is a complex task due to the presence of numerous unpredictable factors influencing human movement. Factors such as varying personal schedules, transportation options, weather conditions, and unexpected events can significantly impact OWs' movements. As a result, developing reliable mobility prediction models that account for these factors becomes a challenging endeavor. Mobility prediction in MCS opens up various application scenarios with far-reaching implications. Firstly, it enables optimal task assignment by accurately anticipating the movement patterns of OWs, ensuring tasks are assigned to workers who are most likely to be present in specific locations at the right time. This enhances data collection efficiency and reduces response time for critical tasks. Secondly, mobility prediction maximizes resource utilization by hiring OWs who may stay at areas where data collection is most needed, ensuring comprehensive coverage of the sensing area. Thirdly, it facilitates quality assurance by allowing proactive measures to cross-validate or supplement data collected from OWs with data from other sources if necessary, ensuring data accuracy and reliability. These scenarios collectively underscore the importance of mobility prediction in enhancing the effectiveness and impact of MCS projects.

A typical flow of an OWs selection scheme consists of two key phases which are discussed below:

In the first phase, worker mobility prediction is carried out to anticipate the future locations of potential workers. This phase leverages historical data, location traces, or other relevant information to predict the movement patterns of workers. By predicting where workers are likely to be at specific times, the system gains insights into their availability and proximity to potential tasks.

Previous approaches for the first phase relied primarily on the assumption that worker mobility follows a Poisson process [13], [28], [29]. For instance, in iCrowd [13], the authors assumed that the probability of a worker being at a specific location at a given time is directly proportional to the number of visit times to that location, following an inhomogeneous Poisson process [30]. Based on this assumption, iCrowd attempted to predict the worker's next location. However, this assumption is not always correct, particularly in urban environments where worker movement can be highly variable. As a result, worker selection based on these predictions may be poor.

In the second phase, OWs selection takes place. During this stage, the system employs various algorithms, such as the maximum utility algorithm, to choose the most suitable workers based on the predicted mobility patterns. The goal is to optimize task allocation and maximize overall utility by selecting workers who can efficiently perform tasks near their anticipated locations.

The maximum utility algorithm is frequently used to select workers like in iCrowd [13]. The algorithm computes each worker's utility based on predicted future locations and selects workers who bring the largest increase in total utility to the group of selected workers in each round until the budget is depleted. However, a limitation of the traditional maximum utility algorithm is its uniform weighting of each location, which proves to be inadequate in scenarios where the number of tasks at each location varies substantially between each sensing cycle (SC).

An alternative approach that considers the number of tasks at each location is presented in DLMV [14]. In the DLMV method, the MCS system selects workers based on a predefined task map, considering the total number of tasks they can complete. However, this approach only accounts for a constant number of tasks over time, thereby failing to address the dynamic nature of task availability in real-world scenarios.

This paper focuses on using OWs for MCS. Previous methods do not take into account the fact that the number of tasks in each region can vary, which can have a significant impact on the amount of sensing data collected. Thus, we propose a new weighted-utility worker selection algorithm that takes into account dynamic changes in task distribution throughout the day. As a result, the proposed algorithm can effectively recruit workers who are well suited to completing a higher number of tasks.

B. HUMAN MOBILITY PREDICTION

Researchers have become interested in the predictability during human mobility. Song et al. [31] found that calculating the entropy of a person's trajectory yielded a high degree of consistency in predicting future destinations. Various practical location prediction models, including early algorithms based on patterns [32], [33], [34], have been developed to this end. For example, WhereNext [35] predicts a moving object's next location using trajectory patterns, whereas geographic temporal semantic location prediction (GTS-LP) [8] determines future positions using pre-defined mobility patterns. These methods, however, are limited by their reliance on pre-established patterns.

Other studies have attempted to predict future locations based on historical knowledge of movements, frequently employing the Markov model [36], [37], [38]. For example, the Mobility Markov Chain (MMC) model was extended to become the n-MMC future location prediction method [36], whereas Wang et al. [37] identified significant locations using an improved density peak-clustering algorithm. Markov-based approaches, on the other hand, struggle to recognize the periodicity and long-term effects of past movements.

Advanced recurrent neural network architectures such as long short-term memory (LSTM) [39] and the gated recurrent unit (GRU) [40] have shown promise in capturing long-term sequential effects and movement patterns. Yao et al. [41] proposed an RNN-based architecture for learning recurrent neural network transition parameters as well as feature embedding. Meanwhile, Gao et al. [42] proposed a latent variable model that uses historical mobility attention to predict a user's future locations. For human location prediction, deep learning methods such as RNNs have proven more effective than pattern-based and Markov-based methods.

In our study, we employ an RNN-based prediction model for workers' future locations. Given deep learning's prowess in handling dynamic datasets, they prove advantageous in MCS systems, offering an effective means of predicting worker movements. While the existing literature presents valuable approaches, key limitations remain. Pattern-based models lack adaptability to evolving mobility patterns, while Markov models struggle with long-term trends. Our proposed work aims to address these gaps by leveraging the strengths of RNN-based methods, providing more accurate and adaptive predictions in dynamic environments.

III. SYSTEM MODEL AND PROBLEM DEFINITION

A. SYSTEM MODEL

This paper addresses the problem of MCS in which a large number of sensing devices are deployed in urban areas to collect data on various phenomena, such as traffic conditions,

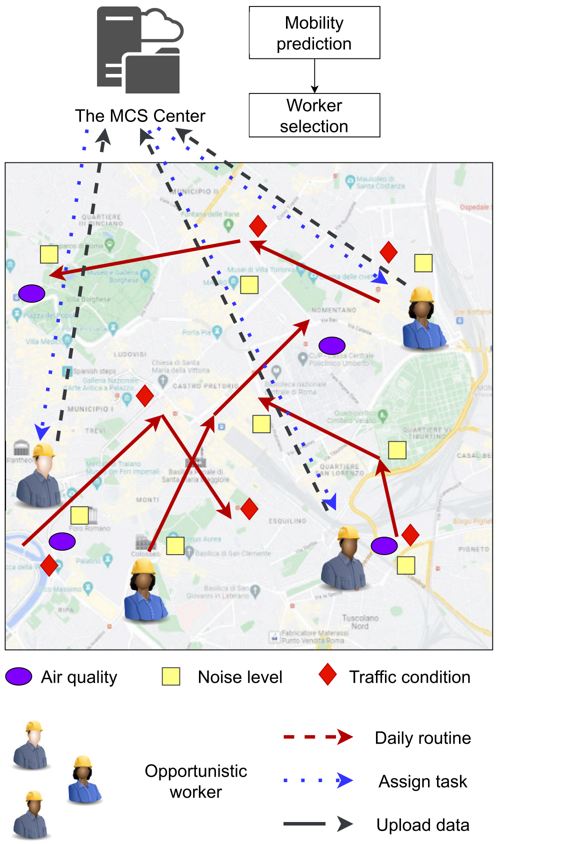


FIGURE 1. The system model for this study.

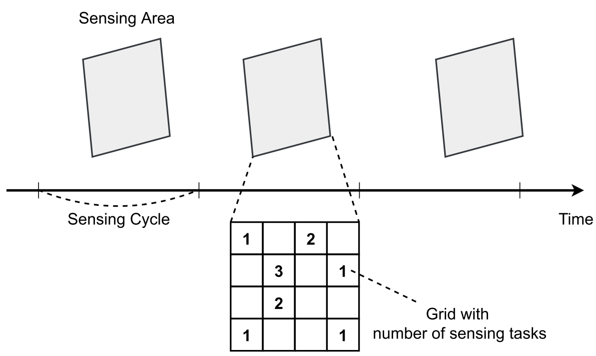


FIGURE 2. Example of a task map.

air quality, and noise levels. At the start of each day, a set of sensing tasks are distributed throughout the city, and the MCS data center must complete as many of these tasks as possible within a limited budget. To do this, the MCS center can either hire participatory workers (e.g., mobile users, taxis) to actively collect data in exchange for high incentives, or recruit OWs for lower incentives. For this paper, we focus on selecting OWs to maximize the number of completed tasks under budget constraints. Figure 1 shows the data collection scheme for the recruited workers.

B. PROBLEM DEFINITION

The sensing area is partitioned into k smaller sections called grids, which are represented as $G = \{g_1, g_2, \ldots, g_k\}$. The

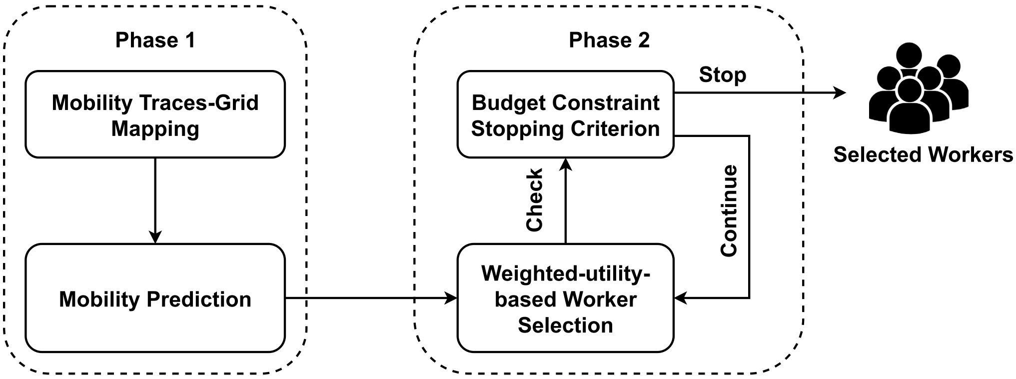


FIGURE 3. The proposed CAMP framework.

working day is divided into m SCs, which are represented as $T = \{t_1, t_2, \ldots, t_m\}$. At the start of each day, the MCS organizer assigns a number of tasks to each grid in every SC. Figure 2 illustrates an example of how tasks are assigned to grids. All tasks that are located in specific grids for the SC can be completed if at least one OW visits the grid during the SC.

In general scenarios, the MCS center will predict workers' next locations and execute the OW selection algorithm for every q SCs. Let $W = \{w_1, w_2, \ldots, w_n\}$ be the set of all workers and W_s is the set of selected workers for q SCs, $W_s \subseteq W$. The budget for every SC is B, and the incentive award given to an OW hired for a SC is I. Thus, the maximum number of workers that the MCS system can hire for the SC is $\frac{B}{I}$.

The objective of this work is to maximize the number of completed tasks, denoted as $|C|$, with a pre-defined budget constraint B. The optimization problem can be formulated as follows:

$$\text{Maximize } |C|$$
$$\text{Subject to: } |W_s| \times I \leq B$$

Terms used in this paper are listed and defined in Table 1.

Comparison to iCrowd [13] and DLMV [14] scheme:

- In iCrowd, it is assumed that there is only one task available at each grid in every sensing cycle (SC). However, in a real scenario, this assumption may not hold true as the number of tasks at each grid can vary significantly throughout the day. In dynamic urban environments, task demands can fluctuate due to changing user needs, events, or time-dependent factors. Ignoring this variability in task distribution could lead to sub-optimal worker selections, resulting in potential inefficiencies and missed opportunities for task completion.
- In DLMV, the authors do consider the differences in the number of tasks at each grid. However, the number of tasks assigned to each grid remains unchanged throughout the day. The dynamic nature of task distribution is not adequately accounted for in their approach. In real-world scenarios, task demands may experience constant fluctuations, influenced by factors such as time of day, traffic patterns, and varying user preferences. Failing to address this dynamism in task allocation could hinder the system's ability to adapt to real-time demands and may result in less efficient worker selection, limiting the overall effectiveness of the MCS process.

TABLE 1. Table of term.

Term	Meaning
MCS	Mobile crowd sensing
OW	Opportunistic worker
RNN	Recurrent neural network
SC	Sensing cycle
B	Budget
G	Set of grids
$\{g_1, g_2, \ldots, g_k\}$	k grids in G
T	Set of sensing cycles
$\{t_1, t_2, \ldots, t_m\}$	m cycles in T
W	Set of workers
$\{w_1, w_2, \ldots, w_n\}$	n workers in W
W_s	Set of selected workers
q	Number of SCs to predict locations
I	Incentive award
$\|C\|$	Completed tasks

IV. METHODOLOGY

A. DESIGN OVERVIEW

In our MCS system, worker selection is centralized and managed by a server. The server collects and stores the mobility history of all volunteers in the target area, and selects workers from this pool for each MCS task. Only the selected workers perform tasks and submit results in each sensing cycle. Our proposed CAMP approach consists of two phases: 1) using historical mobility data, predict each worker's locations; and 2) use the predictions to select workers. This allows the system to effectively recruit a group of workers who are well suited to completing the tasks at hand. The CAMP framework is shown in Fig. 3 and works as follows.

Phase I - data preparation and worker mobility prediction: Using the past movement patterns of workers, predict where

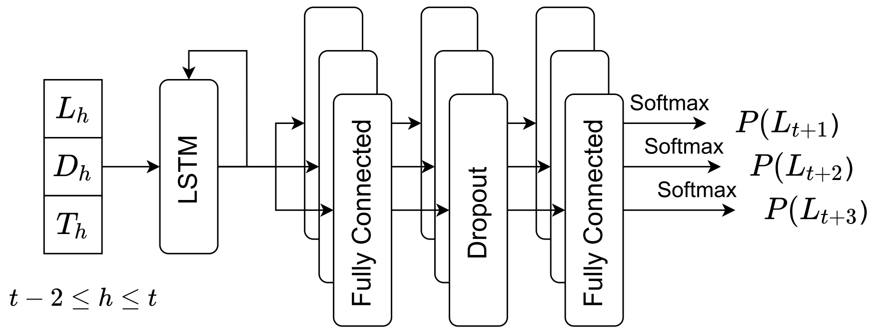


FIGURE 4. The RNN-based multi-output next location prediction model.

the employees will be located during the next SCs. In detail, the next-location prediction model is trained using the mobility traces in two steps:

- Mapping mobility traces - given the historical mobility traces of workers, maps each trace into pre-defined grids.
- Mobility prediction - train the RNN-based multi-output model to predict the next locations of each worker based on their current and historical mobility data.

Phase II - iterative OWs selection: Based on the prediction of each worker's movements, we propose an algorithm to gradually choose OWs.

- The algorithm starts by selecting the worker who has the highest utility among all available workers, and adds her/him to the solution set. The formal definition of utility will be discussed in the next section.
- Then, the algorithm finds the worker who has not been selected and who has the highest incremental utility when combined with the worker(s) already selected. That worker is then added to the selected group.
- The algorithm continues adding workers, one at a time, choosing each worker based on their incremental utility, until the total incentive paid to the selected workers exceeds the pre-defined budget constraint.

B. MULTI-OUTPUT NEXT-LOCATION PREDICTION

The recurrent neural network is a type of neural network that has cycles and internal memory units to process sequential data. LSTM [39] is a commonly used recurrent unit in this type of network. In order to predict a worker's next locations during the next $q = 3$ sensing cycles, the CAMP method uses an RNN-based multi-output model with an LSTM cell. The worker's mobility data are used as input for the model, which then predicts the worker's potential locations in the next time slots $t + k, k \in \{1, 2, 3\}$. The architecture of the location prediction model is illustrated in Fig. 4.

As depicted in Fig. 4, the prediction model includes three layers: the input layer, the recurrent layer, and the output layer.

1) The input layer of the model takes into account the worker's recent locations as well as time-slot and day-of-the-week indices. In Fig. 4, the vectors L_h, D_h, and T_h $(t-2 \leq h \leq t)$ correspond to the location, day of the week, and time slot index of SC h, respectively.
2) The recurrent layer is comprised of an RNN that integrates an LSTM cell.
3) The output layer consists of three parallel branches, each comprising a fully connected layer that uses a ReLU activation function, followed by a dropout layer to counter overfitting, and another fully connected layer that uses softmax activation. Figure 4 depicts the mapping of each branch to predict the workers' whereabouts within a specific time interval. Specifically, the branch predicting the workers' locations during time slot $t + k$, where $k \in 1, 2, 3$, is $P(L_{t+k})$, estimates the probability of visiting every location during that particular time slot.

The cross-entropy loss function is used to train and optimize the model's parameters. The overall loss for the three outputs is calculated as follows:

$$\text{Loss} = \sum^{\text{batch\_size}} \sum_{i=1}^{q} -\log(P(L_{t+i})) \tag{1}$$

where:

- batch_size is the number of samples in a batch.
- q is the number of next time slots to predict worker's location. In our experiment, $q = 3$.
- $P(L_{t+i})$ is the probability of a worker visit locations in time slot $t + i$.

C. WEIGHTED-UTILITY OWs SELECTION

1) Weighted-utility calculation: Given the probability to visit each location during each of the next q SCs, the algorithm iteratively selects the most beneficial OWs for each SC. As mentioned above, priority is assigned to each location depending on the number of tasks at that location at a certain SC. The weighted utility for a set of selected workers, W_s, is calculated as follows:

$$\text{weighted-utility}(W_s) = \sum_{g \in G} \sum_{i=1}^{q} \text{weight}(g) \times \text{prob}_{g,t+i}(W_s) \tag{2}$$

TABLE 2. Experimental setups.

	Values	Default value
Number of days	5	5
Number of workers	$\|N\| = 314$	314
Number of grids	$\|G\| = 100$	100
Number of SCs per day	$\|T\| = 60$	60
Incentive per SC	$I = 1$	1
Budget for a SCs	$B = \{5, 10, 15, 20, 25\}$	25
Number of tasks assigned per SC	$\{5, 10, 15, 20, 25\}$	25

where:

- W_s is the set of selected workers.
- weight(g) is the number of tasks assigned to grid g
- $t+1$ is the consider SC.
- $\text{prob}_{g,t+i}(W_s)$ is the probability of grid g being visited by W_s during SC $t+i$, and is calculated using Eq.(3).

$$\text{prob}_{g,t+i}(W_s) = 1 - \prod_{w \in W_s} (1 - P_{g,t+i}(w)) \tag{3}$$

where:

- w is the worker in W_s
- g is the grid.
- $t+1$ is the consider SC.
- $P_{g,t+i}(w)$ is the probability of worker w visiting grid g during SC $t+i$. Recall that this probability is output by the RNN-based multi-output next-location prediction model.

2) Worker selection: This step involves combining each unselected worker w (i.e., $w \in W \backslash W_s$) with W_s to create a new set ($W_s \cup w$), given W, W_s, and the task map for the current SC. The weighted-utility of each combined set is then calculated as weighted-utility($W_s \cup w$) using Eq.(2). The combined set with the highest weighted-utility is selected as the new W_s for the next iteration. The process continues until the total incentives paid to selected workers exceed the budget constraint. The pseudo-code of the worker selection process is presented in Algorithm 1.

Algorithm 1 Weighted-Utility Opportunistic Worker Selection

Input: Available workers W; budget constraint B.
Output: A set of selected OWs for q sensing cycles.

```
set W_s = ∅
while |W_s| × I ≤ B  and  W ≠ ∅ do
    set MaxUtility = 0, bestW = 0
    for w_j ∈ (W \ W_s) do
        Calculate weighted-utility(W_s ∪ w_j) using Eq.(2)
        if weighted-utility(W_s ∪ w_j) > MaxUtility then
            MaxUtility = weighted-utility(W_s ∪ w_j)
            bestW = w_j
        end if
    end for
    W_s = W_s ∪ bestW
    W = W \ bestW
end while
return W_s
```

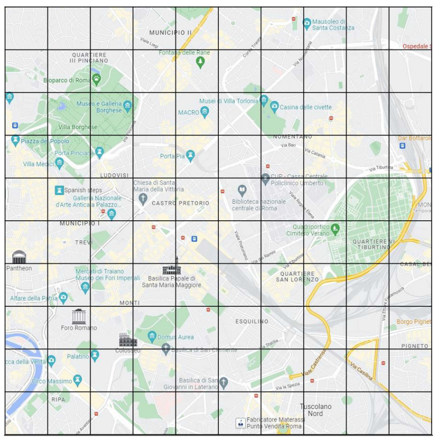


FIGURE 5. Grids map of rome city.

V. EXPERIMENTAL RESULTS

A. BASELINE METHODS

We compared the performance of the CAMP framework with two other approaches: the iCrowd framework [13] and the DLMV framework [14]. iCrowd relies on an inhomogeneous Poisson process for mobility predictions and a greedy maximum utility algorithm for worker selection, whereas the DLMV framework uses a deep learning approach to predict vehicle mobility, and a greedy online approach for task assignment. By comparing the performance of these approaches, we determined the effectiveness of the proposed framework for selecting OWs in MCS.

B. DATASET AND EXPERIMENT SETUPS

To validate the proposed framework, the Crawdad Roma/Taxi dataset was used. The dataset contains the GPS coordinates of approximately 320 taxis collected over the course of 30 days from 2014/02/01 to 2014/03/03. Each trajectory consists of a series of GPS points with timestamps. After data cleaning, a total of 314 taxis were chosen as workers for our scenario. The first 25 days of data were used to train the prediction model, and the last 5 days were used to evaluate the OWs selection algorithm.

TABLE 3. Prediction accuracy (%) from different models.

Procedure	Prediction model	SC $t+1$	SC $t+2$	SC $t+3$	Average
Validation	RNN-based multi-output next-location prediction in CAMP	30.25	18.32	12.31	20.29
	Encoder-decoder RNN model in DLMV [14]	29.54	18.13	12.45	20.04
Testing	RNN-based multi-output next-location prediction in CAMP	26.55	15.33	10.15	17.34
	Encoder-decoder RNN model in DLMV [14]	26.22	15.02	10.16	17.13
	The statistic model in iCrowd [13]	-	-	-	3.6

A $5km \times 5km$ square from Rome's urban area was selected for the experiment. The chosen area was then divided into 10×10 grids, at $500m \times 500m$ each, as shown in Fig. 5, and each grid was assigned an ID. A working day was from 8:00 to 18:00 divided into 60 SCs, with each SC lasting 10 minutes. We varied the number of tasks generated per SC from 5 tasks to 25 tasks. The budget, B, was set to 5, 10, 15, 20, and 25, while keeping the incentive constant at $I = 1$. The experiment setups are summarized in Table 2.

At the beginning of each day, the MCS center assigns a number of tasks randomly to $|G|$ grids throughout the working day. This allows the MCS center to determine the number of tasks at in each grid (i.e., to make the task map) during any given SC, and to recruit a suitable set of OWs. Specifically, a the task map is created that represents the number of tasks assigned to each grid during a particular SC. The map is assigned at the beginning of each day by the MCS center. In this paper, we consider two scenarios:

- Scenario 1: In this scenario, the MCS center maintains the same task map for all SCs. This approach is more suitable for applications where the tasks exhibit relative stability and remain static throughout the day. For instance, consider a scenario where the tasks involve monitoring the air quality in a city. In such cases, the number of tasks required in each grid may remain consistent over time, enabling the MCS center to use a fixed task map for all SCs.
- Scenario 2: In contrast, this scenario involves the MCS center changing the task map for every single sensing cycle (SC). Such an approach is more applicable to applications with dynamic task demands that fluctuate over time. For example, when monitoring traffic conditions in a city, the number of tasks required in each grid may vary throughout the day based on traffic patterns.

By considering these two scenarios, we gain valuable insights into the practical implications of task map management in different MCS applications. Tailoring the task map strategy to the specific characteristics of the tasks and their variations over time allows for a more effective utilization of available resources and improved overall system performance.

The experiments were conducted on a computer equipped with a four-core Intel Xeon W-2123 CPU, 32 GB memory, and a Titan-XP GPU. To implement CAMP and other baseline methods, we utilized the TensorFlow Keras library running on a 64-bit Python 3.8 environment.

C. LOCATION PREDICTION

In this subsection, we evaluate the performance of the proposed location prediction model and compare it to existing approaches in the literature. The proposed RNN-based multi-output location prediction model and the prediction model in DLMV were evaluated on their performance using both validation and test datasets. Please note that the statistics-based models in iCrowd were only evaluated on the test dataset by identifying the location with the highest probability of being visited as the worker's next location. Table 3 shows the prediction accuracy of the models when predicting worker locations in the next SCs (SC $t+1$, SC $t+2$, and SC $t+3$). RNN-based prediction models (CAMP and DLMV) were found more effective in predicting worker locations than statistics-based models. In particular, these models have the highest accuracy when predicting worker locations in the next SC, with accuracy slightly decreasing for predictions further in the future. This suggests that a worker's movement history has a more significant impact on their locations in the near term than in the long term. These results demonstrate the effectiveness of using RNN-based models for predicting worker locations in MCS, and emphasize the significance of considering a worker's recent movement patterns when making predictions about their future locations.

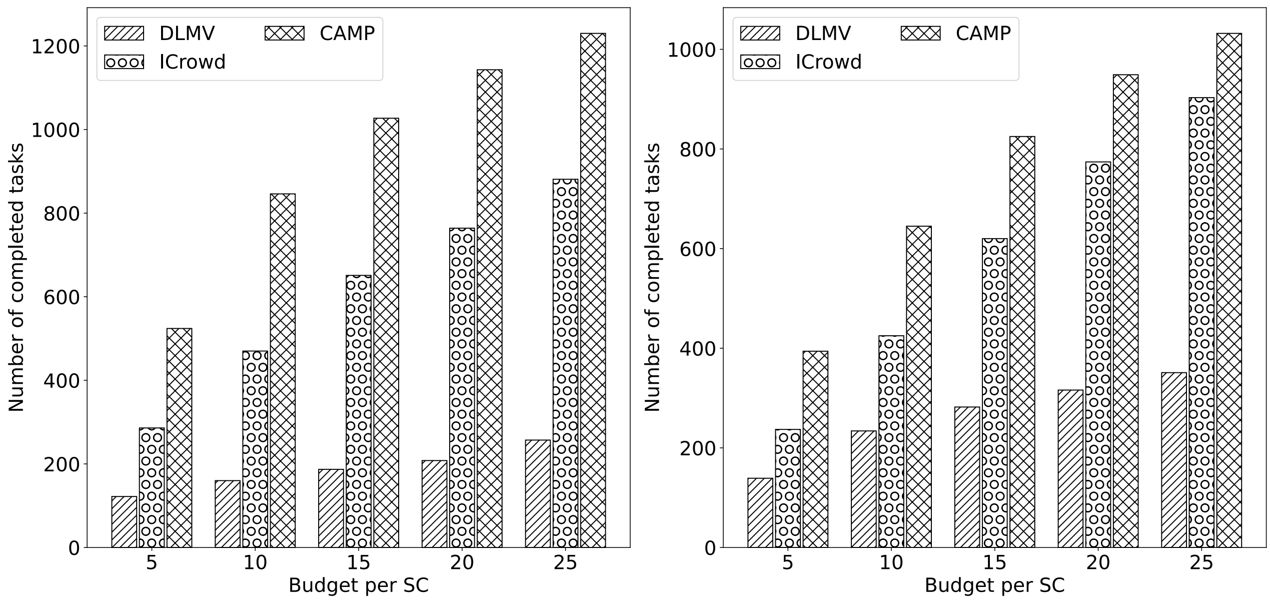
D. EXPERIMENT RESULTS

1) DIFFERENT BUDGET VALUES

This subsection discusses the performance of the CAMP framework under different budget constraints and scenarios. The relationship between the budget allocated for each sensing cycle and the number of completed sensing tasks is analyzed. Figure 6 presents a comparison of performance under varying budget constraints, where the budget for each SC ranges from 5 to 25 and the number of tasks generated per SC is set to the default value of 25. Please refer to the following calculation:

- total budget = budget per SC $|B| \times$ number of SC per day $|T| \times$ number of test days. Thus in this experiment, the total budget varies from 1500 (e.g., $5 \times 60 \times 5$) to 7500 (e.g., $25 \times 60 \times 5$).
- total number of tasks = number of tasks per SC $\times$ number of SC per day $|T| \times$ number of test days. Thus in this experiment, the total number of tasks is 7500 tasks (e.g., $25 \times 60 \times 5$)

As the budget increases, the number of completed tasks generally increases as more workers are recruited to complete



(a) Scenario 1: Same task map for all SCs

(b) Scenario 2: Different task map every SC

FIGURE 6. The performance comparison in different budgets. In this experiment, the budget for each SC is varied from 5 to 25 while the number of tasks generated every SC is set to 25 tasks per SC. The y-axis shows the total number of completed tasks over 5 test days.

the tasks. CAMP consistently outperforms both ICrowd and DLMV across all budget in terms of the number of completed tasks. In Scenario 1, CAMP achieves an average of 56.3 percent more completed tasks than ICrowd, showing a significant improvement in task completion rates. Similarly, in the same scenario, CAMP outperforms DLMV by an impressive 410.7 percent, clearly demonstrating the remarkable impact of our proposed framework.

Similarly, in Scenario 2, CAMP continues to demonstrate its superiority, achieving an average of 29.9 percent more completed tasks than ICrowd. Moreover, in the same scenario, CAMP outperforms DLMV by 190.8 percent. This improvement reaffirms the consistent effectiveness of the CAMP approach in dynamic task environments.

One noteworthy finding is that the CAMP framework outperformed the iCrowd method. This can be attributed to two key reasons. Firstly, the iCrowd method assumes workers' mobility follows a specific statistical process, which may not accurately capture the complexities of human mobility patterns. In contrast, the CAMP framework utilizes an RNN-based multi-output prediction model, which has the advantage of more accurately predicting workers' future locations based on their past mobility information. This accuracy in mobility prediction results in better worker selection. Secondly, the iCrowd framework does not consider the number of tasks at different locations when selecting workers. In contrast, the CAMP framework employs a weighted-utility worker selection algorithm that assigns higher weights to workers who are more likely to visit locations with a larger number of tasks. This strategic approach ensures more efficient worker selection and distribution of tasks, leading to higher task completion rates.

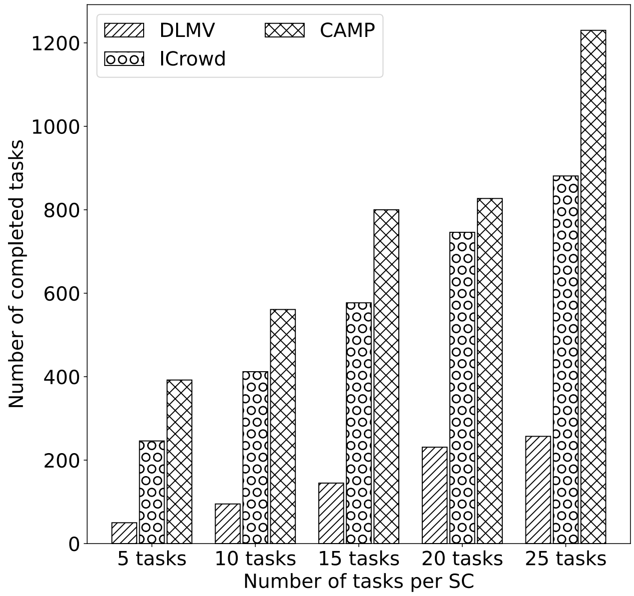
Table 3 provides further insights into the prediction performance of the DLMV and CAMP methods. The prediction accuracy of both methods was found to be similar. However, the number of completed tasks using the DLMV method was significantly lower. This discrepancy can be attributed to the DLMV approach's reliance on a worker selection algorithm that solely considers the likelihood of workers visiting specific locations. In contrast, both the CAMP framework and the iCrowd approach take into account a worker's likelihood of visiting all locations when calculating utility, resulting in improved performance and a higher number of completed tasks.

Overall, the technical analysis of the results underscores the superiority of the CAMP framework in accurately predicting worker mobility, efficiently selecting workers based on task distribution, and ultimately enhancing task completion rates in MCS scenarios.

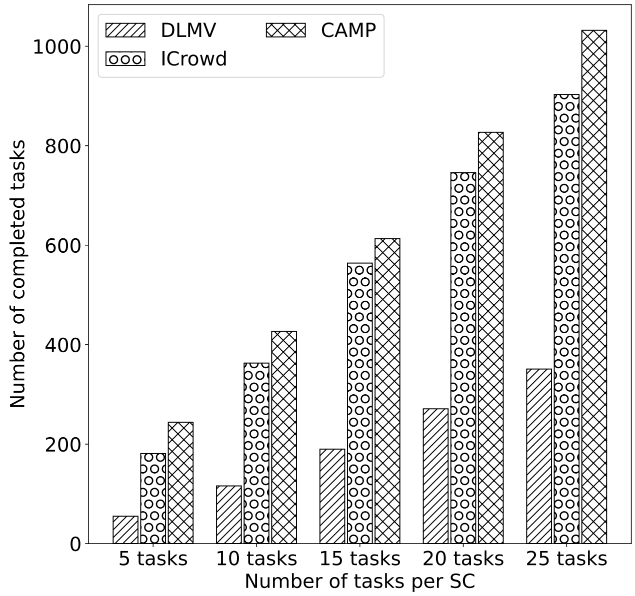
2) DIFFERENT NUMBERS OF TASKS

This subsection extends the performance evaluation of the CAMP framework, providing in-depth analysis and insights into its superiority over baseline methods under different task load scenarios. The evaluation considered a range of tasks varying from 5 tasks per sensing cycle (SC) to 25 tasks per SC, enabling a comprehensive assessment of the framework's capabilities. The budget for each SC is set to 25. Please refer to the following calculation:

- total budget = budget per SC $|B|$ $\times$ number of SC per day $|T|$ $\times$ number of test days. Thus in this experiment, the total budget is 7500 (e.g., $25 \times 60 \times 5$).
- total number of tasks = number of tasks per SC $\times$ number of SC per day $|T|$ $\times$ number of test days. Thus



(a) Scenario 1: Same task map for all SCs



(b) Scenario 2: Different task map every SC

FIGURE 7. Performance comparison of different numbers of tasks generated for every SC. In this experiment, the number of tasks generated every SC is varied from 5 tasks per SC to 25 tasks per SC while the budget for each SC is set to 25. The y-axis shows the total number of completed tasks over 5 test days.

in this experiment, the total number of tasks varies from 1500 tasks (e.g., $5 \times 60 \times 5$) to 7500 tasks (e.g., $25 \times 60 \times 5$)

As illustrated in Figure 7, CAMP consistently exhibits a remarkable performance advantage over both ICrowd and DLMV across all task settings. In Scenario 1, CAMP achieves an average of 33.1 percent and 389.7 percent more completed tasks compared to ICrowd and DLMV, respectively. Similarly, in Scenario 2, CAMP achieves an average of 14.0 percent and 219.7 percent more completed tasks compared to ICrowd and DLMV, respectively.

The performance of baseline methods showed some improvement as the number of tasks increased. However, this improvement was not as substantial as that observed with the proposed CAMP framework. The key differentiating factor lies in CAMP's ability to prioritize and select workers efficiently based on their likelihood of visiting locations with a significant number of tasks. By making more reasonable decisions in worker selection, CAMP optimizes task allocation and enhances overall efficiency in task completion.

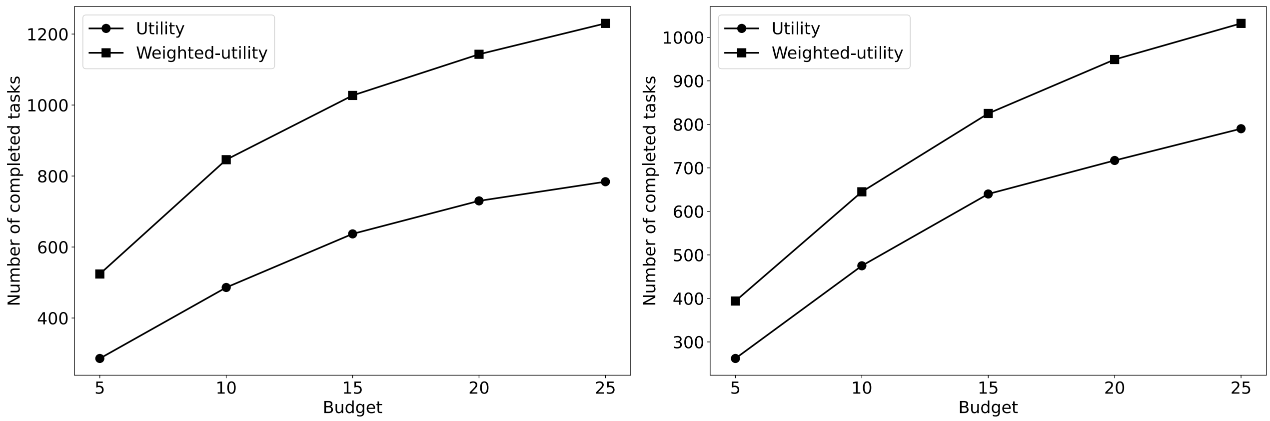
Moreover, the utilization of an RNN-based multi-output prediction model in the CAMP framework enables accurate prediction of workers' movements. This predictive capability plays a crucial role in dynamically adapting to changing task loads, allowing CAMP to anticipate worker movement patterns and proactively assign them to locations where their expertise is most needed. Consequently, task allocation is improved, leading to enhanced overall performance.

The presented results serve as compelling evidence of the CAMP framework's adaptability to varying task loads and reinforce its superiority over baseline methods. By effectively addressing the challenges posed by fluctuating workloads, the CAMP framework emerges as a robust solution that outperforms traditional approaches. These findings highlight the practical value and potential impact of integrating the CAMP framework into real-world scenarios, where dynamic task assignment and efficient resource utilization are critical considerations.

E. EFFECTIVENESS OF THE WEIGHTED-UTILITY ALGORITHM

In this subsection, we evaluate the effectiveness of using a weighted-utility algorithm for OWs selection in Phase 2 of the CAMP framework. The goal of this evaluation is to determine whether the weighted-utility approach improves the overall performance and efficiency of the worker selection process.

Figure 8 compares how well the weighted-utility method performed under different budget constraints and scenarios. The results consistently show that the weighted-utility method outperformed the utility-based method in all budget scenarios. In scenarios 1, the weighted-utility approach achieves an average of 63.2 percent more completed tasks, showcasing its superior performance. Similarly, in scenarios 2, the weighted-utility algorithm outperforms the utility-based algorithm by 33.3 percent, further confirming its effectiveness. The main advantage of the weighted-utility approach is its ability to accurately prioritize locations based on expected task load. This means the MCS system can select workers who are more likely to visit high-priority locations, resulting in more completed tasks within the given budget. By using resources more efficiently, the system achieves higher task completion rates and overall performance. This evaluation clearly demonstrates the potential of the weighted-utility algorithm to revolutionize the worker



(a) Scenario 1: Same task map for all SCs

(b) Scenario 2: Different task map every SC

FIGURE 8. Effectiveness of the weighted-utility worker selection algorithm. In this experiment, the budget for each SC is varied from 5 to 25 while the number of tasks generated every SC is set to 25 tasks per SC. The y-axis shows the total number of completed tasks over 5 test days.

selection process in the CAMP framework. Adopting this approach in MCS systems can lead to improved task completion rates, better resource management, and overall system efficiency, making it a promising advancement in worker selection strategies for MCS applications.

VI. CONCLUDING REMARKS

This paper presents a two-phase framework named context-aware worker recruitment based on a mobility prediction model (CAMP) which aimed at tackling the problem of selecting OWs in MCS. The main innovation of the CAMP framework centers on its integration of a precise multi-output RNN model for predicting worker mobility and a distinctive weighted-utility worker selection algorithm. This algorithm takes into account the varying task distribution across different locations and time slots, resulting in more accurate and efficient worker selection. The evaluation conducted with the real-world Crawdad Roma/Taxi dataset clearly illustrates the superiority of the CAMP framework over baseline methods in terms of the number of completed tasks. Notably, CAMP outperforms both iCrowd and DLMV in both considered scenarios (i.e., using the same task map and changing task map), with the variation in the budget and the number of tasks generated in each sensing cycle (SC). This performance showcases the substantial potential of CAMP to significantly enhance worker selection efficiency in MCS applications.

However, one limitation of the proposed framework is its lack of consideration for other influential factors that could affect worker selection, such as worker availability, skill levels, or past performance. To address this limitation, future work may involve incorporating these additional factors into the worker selection process. For instance, extending the weighted-utility worker selection algorithm to prioritize more available workers could potentially enhance task completion rates, especially within specific time constraints.

Furthermore, evaluating the CAMP framework using different real-world datasets and diverse MCS scenarios would offer valuable insights into its performance and applicability across various settings. By testing the proposed method in different contexts, we can gain a more comprehensive understanding of its strengths and potential areas for improvement, leading to a more robust and effective worker selection approach in practical MCS implementations.

In conclusion, the CAMP framework presents a novel and effective solution to optimize OWs selection in MCS. While acknowledging its limitation and suggesting potential improvements, the contributions of CAMP open a new direction for future advancements in worker selection strategies for MCS applications.

REFERENCES

[1] H. Wang, "A survey of application and key techniques for mobile crowdsensing," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–11, Nov. 2022.

[2] S. Zhao, G. Qi, T. He, J. Chen, Z. Liu, and K. Wei, "A survey of sparse mobile crowdsensing: Developments and opportunities," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 73–85, 2022.

[3] D. Cicek and B. Kantarci, "Use of mobile crowdsensing in disaster management: A systematic review, challenges, and open issues," *Sensors*, vol. 23, no. 3, p. 1699, Feb. 2023.

[4] H. El Hafyani et al., "Learning the microenvironment from rich trajectories in the context of mobile crowd sensing," *Geoinformatica*, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s10707-022-00471-4, doi: 10.1007/s10707-022-00471-4.

[5] Y. Chen, W. Zhao, and C. Xu, "A survey on task allocation in mobile crowd sensing: Current state and challenges," in *Proc. IEEE 6th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Beijing, China, Oct. 2022, pp. 220–225.

[6] T. Kandappu, A. Misra, and R. Tandriansyah, "Collaboration trumps homophily in urban mobile crowdsourcing," in *Proc. ACM Conf. Comput. Supported Cooperat. Work Social Comput.*, Portland, OR, USA, Feb. 2017, pp. 902–915.

[7] T. Kandappu, A. Misra, S.-F. Cheng, N. Jaiman, R. Tandriansyah, C. Chen, H. C. Lau, D. Chander, and K. Dasgupta, "Campus-scale mobile crowd-tasking: Deployment & behavioral insights," in *Proc. 19th ACM Conf. Comput.-Supported Cooperat. Work Social Comput.*, San Francisco, CA, USA, Feb. 2016, pp. 800–812.

[8] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 1–33, Dec. 2013.

[9] P. Cheng, X. Lian, L. Chen, and C. Shahabi, "Prediction-based task assignment in spatial crowdsourcing," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, San Diego, CA, USA, Apr. 2017, pp. 997–1008.

[10] Q. T. Ngo, D. T. Lan, S. Yoon, W.-S. Jung, T. Yoon, and D. Yoo, "Companion mobility to assist in future human location prediction," *IEEE Access*, vol. 10, pp. 68111–68125, 2022.

[11] B. Altaf, L. Yu, and X. Zhang, "Spatio-temporal attention based recurrent neural network for next location prediction," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 937–942.

[12] S. Alemany, J. Beltran, A. Perez, and S. Ganzfried, "Predicting hurricane trajectories using a recurrent neural network," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, 2019, vol. 33, no. 1, pp. 468–475.

[13] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "ICrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.

[14] X. Zhu, Y. Luo, A. Liu, W. Tang, and M. Z. A. Bhuiyan, "A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4648–4659, Jul. 2021.

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

[16] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, D. Zhang, B. Guo, and Q. Lv, "HyTasker: Hybrid task allocation in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 598–611, Mar. 2020.

[17] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Orlando, FL, USA, Nov. 2013, pp. 324–333.

[18] D. Deng, C. Shahabi, U. Demiryurek, and L. Zhu, "Task selection in spatial crowdsourcing from worker's perspective," *GeoInformatica*, vol. 20, no. 3, pp. 529–568, Jul. 2016.

[19] Z. Wang, J. Zhao, J. Hu, T. Zhu, Q. Wang, J. Ren, and C. Li, "Towards personalized task-oriented worker recruitment in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 2080–2093, May 2021.

[20] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Trans. Spatial Algorithms Syst.*, vol. 1, no. 1, pp. 1–28, Aug. 2015.

[21] X. Chen, "A stable task assignment scheme in crowdsourcing," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE), IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, New York, NY, USA, Aug. 2019, pp. 489–494.

[22] H. To, "Task assignment in spatial crowdsourcing: Challenges and approaches," in *Proc. 3rd ACM SIGSPATIAL PhD Symp.*, Burlingame, CA, USA, Oct. 2016, pp. 1–4.

[23] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3747–3757, Oct. 2018.

[24] F. Yucel, M. Yuksel, and E. Bulut, "Coverage-aware stable task assignment in opportunistic mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3831–3845, Apr. 2021.

[25] F. Yucel and E. Bulut, "Online stable task assignment in opportunistic mobile crowdsensing with uncertain trajectories," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9086–9101, Jun. 2022.

[26] J. Wang, J. Liu, and G. Zhao, "Two-phased participant selection method based on partial transfer learning in mobile crowdsensing," *ACM Trans. Sensor Netw.*, vol. 19, no. 2, pp. 1–17, May 2023.

[27] J. Zhang and X. Zhang, "Multi-task allocation in mobile crowd sensing with mobility prediction," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 1081–1094, Feb. 2023.

[28] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "TaskMe: Multi-task allocation in mobile crowd sensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Heidelberg, Germany, Sep. 2016, pp. 403–414.

[29] D. Zhang, H. Xiong, L. Wang, and G. Chen, "CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Seattle, WA, USA, Sep. 2014, pp. 703–714.

[30] J. Weinberg, L. D. Brown, and J. R. Stroud, "Bayesian forecasting of an inhomogeneous Poisson process with applications to call center data," *J. Amer. Stat. Assoc.*, vol. 102, no. 480, pp. 1185–1198, Dec. 2007.

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

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

[33] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 194–200.

[34] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "GMove: Group-level mobility modeling using geo-tagged social media," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1305–1314.

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

[36] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proc. 1st Workshop Meas., Privacy, Mobility*, Bern, Switzerland, Apr. 2012, pp. 1–6.

[37] H. Wang, Z. Yang, and Y. Shi, "Next location prediction based on an AdaBoost-Markov model of mobile users," *Sensors*, vol. 19, no. 6, p. 1475, Mar. 2019.

[38] M. Chen, Y. Liu, and X. Yu, "NLPMM: A next location predictor with Markov modeling," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Tainan, Taiwan: Springer, 2014, pp. 186–197.

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

[40] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[41] D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, Nov. 2017, pp. 2411–2414.

[42] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. World Wide Web Conf.*, San Francisco, CA, USA, May 2019, pp. 2750–2756.

QUAN T. NGO received the B.S. degree in electronics and telecommunications engineering from The University of Danang, University of Science and Technology, Da Nang, Vietnam, in 2016. He is currently pursuing the M.S./Ph.D. degree with the Department of Electrical, Electronic and Computer Engineering, University of Ulsan, South Korea. His research interests include computer vision, human mobility prediction, and ML-based stock movement prediction.

SEOKHOON YOON (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science and engineering from The State University of New York at Buffalo (SUNY Buffalo), in 2005 and 2009, respectively. After receiving the Ph.D., he was a Senior Research Engineer in the defense industry, where he designed several tactical wireless network solutions. He is currently a Professor with the University of Ulsan, South Korea, where he leads the Advanced Mobile Networks and Intelligent Systems Laboratory. His research interests include opportunistic networking, human mobility, intelligence defined networking, and machine learning based IoT services.