



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Doctor of Philosophy

**Human Mobility Prediction and
Application in Mobile Crowd Sensing**

**The Graduate School of the University of Ulsan
Department of Electrical, Electronic and Computer Engineering**

NGO TAN QUAN

**Human Mobility Prediction and
Application in Mobile Crowd Sensing**

Supervisor: Prof. Seokhoon Yoon

A Dissertation

**Submitted to
the Graduate School of the University of Ulsan
In partial Fulfillment of the Requirements
for the Degree of**

Doctor of Philosophy

by

Ngo Tan Quan

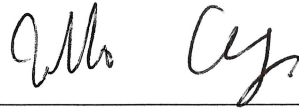
Department of Electrical, Electronic and Computer Engineering

University of Ulsan, Korea

December 2023

Human Mobility Prediction and Application in Mobile Crowd Sensing

This certifies that the dissertation of Ngo Tan Quan is approved by:



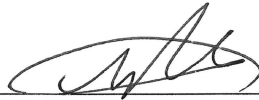
Committee Chair: Prof. Jin-Ho Chung



Committee Member: Prof. Daehwan Kim



Committee Member: Prof. DaeHan Ahn



Committee Member: Dr. Woo-Sung Jung



Committee Member: Prof. Seokhoon Yoon, *Adviser*

Department of Electrical, Electronic and Computer Engineering

University of Ulsan, Korea

December 2023

Dedicated to

Dedicated to my family, friends, and all those who have supported me
along this journey...

...and, in special appreciation, to my beloved wife and my brave son.

Acknowledgments

First and foremost, I would like to express my heartfelt gratitude to my advisor, Prof. Seokhoon Yoon of the Department of Electrical, Electronic and Computer Engineering at the University of Ulsan. His exceptional guidance and support have been instrumental in shaping my academic path. Through his critical feedback and insightful suggestions, I have not only honed my research skills but also acquired a wealth of invaluable knowledge. Without his mentorship, none of my achievements would have been possible.

I extend my sincere appreciation to the committee members who diligently reviewed my thesis and provided invaluable feedback. Their expertise and constructive comments significantly contributed to the enhancement of my thesis's quality.

My profound thanks go out to my parents, parents-in-law, sisters, brothers, and friends for their constant encouragement and boundless inspiration during my academic journey. Their support has been a constant source of motivation.

I am also grateful to my esteemed colleagues at the Advanced Mobile Networks and Intelligence Systems Lab, whose valuable insights and shared memories have enriched my research experience.

Last but certainly not least, I owe an immeasurable debt of gratitude to my beloved wife and my son. Their enduring love, boundless patience, and unwavering support have been the cornerstone of my success during my time at the University of Ulsan.

Abstract

Human mobility prediction plays a pivotal role in a wide range of applications, including targeted advertising, traffic management, urban planning, and the containment of contagious diseases. In the context of Mobile Crowd Sensing (MCS), where mobile users collect sensory data in their daily lives, human mobility prediction offers the potential to enhance the data-collection process by optimizing resource allocation based on future human locations. However, predicting human mobility is a challenging problem, characterized by non-linear, dynamic patterns influenced by various factors such as time, weather, and events. Therefore, the main goal of this dissertation is to address two primary challenges: the development of accurate and efficient human mobility prediction models, and the effective application of human mobility prediction to enhance MCS systems. To achieve these goals, we propose two distinct frameworks.

In the first framework, we focus on location prediction within the realm of human mobility. Prior research primarily concentrated on modeling individual mobility patterns in isolation, often neglecting valuable insights that can be gained from the movements of companions. To bridge this gap, our research introduces a two-phase framework that capitalizes on spatio-temporal contexts embedded in both the person of interest (POI) and their companions' mobility patterns. Initially, the framework identifies companions using spatial closeness (SC) and person ID embedding (PIE) vectors. Additionally, it effectively addresses the challenge of dimensionality through the application of a stacked autoencoder, where the encoder part plays a pivotal role in solving the problem. In the second phase, a bidirectional recurrent neural network (BRNN)-based multi-output model is employed to precisely forecast the POI's future locations across various time slots. This process is optimized using a weighted loss approach that takes into account the significance of each future time slot for accurate location predictions.

In the second framework, our focus shifts to the selection of opportunistic workers (OWs) in the context of Mobile Crowd Sensing (MCS). We propose a novel framework known as 'Context-Aware Worker Recruitment based on a Mobility Prediction Model' (CAMP) to efficiently choose the most suitable workers who can complete a maximum number of sensing tasks within a given budget constraint. CAMP operates in two phases. Initially, it predicts the future locations of workers and subsequently selects a group of workers to carry out sensing tasks. Diverging from traditional approaches, CAMP enhances worker

mobility prediction using a recurrent neural network-based model, substantially improving location prediction accuracy during the first phase. In the second phase, we introduce a weighted-utility algorithm that takes into account variations in task distribution across different locations and sensing cycles. The integration of accurate mobility prediction and the weighted-utility worker selection algorithm positions CAMP as a superior solution for optimizing task allocation within the MCS framework.

We conducted comprehensive experiments to assess the performance of the proposed next-location prediction and worker selection frameworks across diverse large datasets. The evaluation results demonstrate that the location prediction frameworks achieve higher predictive accuracy, while the worker selection framework successfully completes more tasks within the same budget constraints compared to their alternatives.

Contents

Acknowledgments	1
Abstract	2
1 Introduction	9
1.1 Human Mobility Prediction	9
1.2 The Proposed Companion Mobility to Assist in Future Human Location Prediction Framework	9
1.3 Opportunistic Worker Selection In Mobile Crowd Sensing	11
1.4 The Proposed Context-Aware Worker Recruitment for MobileCrowd Sensing Based on Mobility Prediction Framework	12
1.5 Dissertation Organization	13
2 Background and Related Works	14
2.1 Human Mobility Analysis	14
2.2 Location Prediction	14
2.3 Social Correlation Detection	16
2.4 Tasks Allocation in Mobile Crowd Sensing	17
2.5 Opportunistic Worker Selection in Mobile Crowd Sensing Using Human Mobility Prediction	18
3 Companion Mobility to Assist in Future Human Location Prediction	21
3.1 Problem Definition	21
3.2 Methodology	23
3.2.1 Companion Selection Methods	24
3.2.2 Location-Day-Time slot-ID Autoencoder (LDTI-AE)	27
3.2.3 BRNN-based Future Location Prediction Model	28

3.2.4	Parameter Learning	31
3.3	Datasets	31
3.4	Experiment Results	34
3.4.1	Baseline Models	34
3.4.2	Implementation Details	35
3.4.3	Experiment Results	37
3.4.4	Sensitivity of Hyper-parameters	38
3.4.5	Effectiveness of Weighted Loss	40
3.4.6	Person ID Embedding Companion Selection Validation	42
3.5	Chapter Summary	45
4	Context-Aware Worker Recruitment for MobileCrowd Sensing Based on Mobility Prediction	47
4.1	System Model and Problem Definition	47
4.1.1	System Model	47
4.1.2	Problem Definition	49
4.2	Methodology	51
4.2.1	Design Overview	51
4.2.2	Multi-output Next-location Prediction	52
4.2.3	Weighted-utility OWs Selection	53
4.3	Experimental Results	55
4.3.1	Baseline Methods	55
4.3.2	Dataset and Experiment Setups	56
4.3.3	Location Prediction Performance	59
4.3.4	Experiment Results	60
4.3.5	Effectiveness of the Prediction model	65
4.3.6	Effectiveness of the Weighted-utility Algorithm	67
4.4	Chapter Summary	69
5	Concluding Remarks	70
5.1	Summary of the Contribution	70
5.2	Future Works	72
5.2.1	Enhancing Human Mobility Prediction	72
5.2.2	Expanding Worker Selection Framework	73

List of Figures

3.1	A toy example of how the prediction model work with $k = 3, L = \{A, B, C\}, m \in \{1, 2, 3\}$	22
3.2	Two-phase framework for predicting an individual’s future locations.	23
3.3	Model architecture for learning a PIE matrix in phase 1.	26
3.4	LDTI - Autoencoder architecture in phase 1.	28
3.5	The BRNN-based future locations prediction model.	29
3.6	Input-Label timelines of five test sets.	33
3.7	Top-1 accuracies on the two datasets from the different methods predicting POI locations in several time slots.	36
3.8	Performance with varying parameters on Dartmouth dataset. (a) Person ID embedding size. (b) Location embedding size. (c) Bi-LSTM hidden unit size. (d) The first fully connected layer size.	39
3.9	Performance comparison of the proposed model with PIE, trained using normal loss and weighted loss.	41
3.10	Synthetic data construction for five new virtual people with IDs 51, 52, 53, and 54.	42
3.11	t-SNE plot of the PIE vectors for all 54 people. Each point in the plot indicates a PIE vector.	44
4.1	The system model for this study.	48
4.2	Example of a task map.	49
4.3	The proposed CAMP framework.	51
4.4	The RNN-based multi-output next location prediction model.	52
4.5	Grids map of Rome city.	57

4.6	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.	61
4.7	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.	64
4.8	Effectiveness of the RNN-based model for worker selection. 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.	66
4.9	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.	68

List of Tables

2.1	Comparison of some location prediction approaches.	15
3.1	Table of term for location prediction framework	22
3.2	Samples of location-person vectors.	24
3.3	Statistics on the two experimental datasets after pre-processing.	32
3.4	Average top-1, top-3, and top-5 accuracies with Dartmouth data.	38
3.5	Average top-1, top-3, and top-5 accuracies with UB data.	38
3.6	Number of parameters of different models on the two datasets.	38
3.7	The cosine similarity scores between different people's PIE vectors.	43
4.1	Table of term for OW selection framework	50
4.2	Experimental setups	57
4.3	Prediction accuracy (%) from different models	59

Chapter 1

Introduction

1.1 Human Mobility Prediction

Human mobility prediction serves a critical role in a wide spectrum of applications, encompassing fields like advertising, traffic management, urban planning, and the control of contagious diseases [1–3]. For instance, governments can enhance transportation planning and alleviate traffic congestion and crowd management by predicting people’s whereabouts [4]. Ride-hailing services such as Uber and Grab rely heavily on precise mobility prediction algorithms to accurately gauge customer travel demands and allocate resources efficiently [5, 6]. In opportunistic mobile social networks, where each mobile user is regarded as a node, accurate prediction of the next node’s position reduces the number of route discoveries and minimizes hop counts for source-to-destination paths [7, 8]. Notably, in industrial zones, predicting worker locations based on their recent movements is vital for improving safety and expediting rescue operations in the event of an incident [9, 10].

Existing methods, such as [11–14], primarily concentrate on developing models that capture the sequential nature of human mobility preferences for the person-of-interest (POI) only. These methods often overlook the spatio-temporal context of a person’s movement companions, thus missing out on valuable information associated with companion mobility.

1.2 The Proposed Companion Mobility to Assist in Future Human Location Prediction Framework

Motivated by the fact that people frequently move with friends, colleagues, or coworkers; therefore, incorporating mobility information about companions can improve the performance of a location-prediction model, we propose a two-phase framework that takes ad-

vantage of a person’s companions’ mobility information to predict the next location of the person-of-interest (POI). In a nutshell, we propose two new methods to choose companions of the POI. Then, utilizing mobility information of both the POI and his/her chosen companions, a bidirectional recurrent neural network (BRNN) predicts the POI’s future locations.

For the first phase, two companion selection methods are proposed to determine the POI’s companions so their mobility information can help predict the POI’s future locations. The first method uses a spatial closeness (SC) [15] metric that measures the similarity in mobility between the POI and others, and then selects the most similar individuals as the POI’s companions. For the second companion selection method, a person ID embedding (PIE) matrix is learned. Each embedding vector in the PIE matrix represents an individual’s mobility characteristics. Similarity scores between the POI’s PIE vector and vectors of other people are calculated using cosine similarity. Those who have PIE vectors with the highest similarity score to the POI’s PIE vector are selected as her/his companions.

In machine learning, one-hot encoding vectors are conventionally used to represent categorical inputs to the model. However, extensive use of one-hot vectors can lead to the curse of dimensionality, especially as vector lengths increase. This can adversely affect model performance and training time. Therefore, in the first phase of the framework, we employ a stacked autoencoder [16] to address the curse of dimensionality. Specifically, instead of directly inputting one-hot vectors of various input features (such as location, day of the week, time slot of the day, and ID) into the prediction model in the second phase, they first pass through the encoder layer of the autoencoder. The encoder layer transforms input features into dense representations, effectively reducing input dimensionality and capturing intricate correlations between input features, thereby improving the model’s prediction.

In the second phase, the BRNN-based multi-output prediction model forecasts future personal locations using recent mobility data from the POI and the selected companions. The advantage of BRNN lies in its ability to process input both in the forward and reverse temporal order, mitigating the limitation of traditional RNNs that only process input sequentially. This enables the BRNN to consider future context. Furthermore, the multi-output model is designed to predict the person’s location in several time slots with varying time gaps. For example, if the time slot length is 15 minutes and the person is currently in time slot t , the model predicts positions for time slots $t + 1, t + 4, t + 8, t + 12$, and $t + 16$ (corresponding to 15 minutes, one hour, two hours, three hours, and four hours later, re-

spectively).

The proposed model was trained and tested using two large-scale Wi-Fi trace datasets, the Dartmouth dataset [17] and the UB dataset [18]. We compared our model's performance to that of several baselines: the Markov model [11], SERM [12], VANext [13], and DPBPT [14]. The results show that the proposed model outperformed the baseline models in predicting the person's next several locations.

1.3 Opportunistic Worker Selection In Mobile Crowd Sensing

In addition to its applications in the aforementioned contexts, human mobility prediction plays a crucial role in the realm of Mobile Crowd Sensing (MCS). MCS, a method characterized by the active participation of dynamically moving mobile users, often referred to as "workers," has emerged as a highly promising approach for collecting a diverse array of urban data. Workers, equipped with mobile devices, contribute to the acquisition of valuable environmental, economic, and social data. This method has garnered significant attention from both academic and industrial communities, primarily owing to its potential to offer highly accurate, cost-effective, and scalable sensing solutions, as noted in previous surveys [19,20].

MCS, however, introduces its unique set of challenges. Workers' mobility patterns can be intricate and unpredictable, with individuals traversing various locations, sometimes intermittently or with varying frequencies. Here, the role of human mobility prediction comes into focus. By leveraging predictive models that estimate the future locations of workers, we can significantly enhance the efficiency and effectiveness of MCS data collection. These models, as presented in this dissertation, provide MCS systems with the ability to anticipate worker locations, consequently optimizing resource allocation and task distribution in real-time. As a result, the integration of human mobility prediction into MCS not only refines the data collection process but also bolsters its capabilities to address various environmental, economic, and social challenges more adeptly, enabling the development of innovative applications that benefit society at large [21,22].

Task allocation in MCS involves strategically assigning sensing tasks to mobile workers to efficiently collect data in urban areas [23]. 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 [24,25]. 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.

Opportunistic workers (OWs) play an important role in MCS [26,27]. 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.

1.4 The Proposed Context-Aware Worker Recruitment for MobileCrowd Sensing Based on Mobility Prediction Framework

To address the OW selection problem in MCS, we propose 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 [28–30]. 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 [31] ignores variations in the number of tasks at each location or DLMV [32] 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 [33]. The results show that the proposed framework can outperform baseline models in the number of completed tasks for the same budget amount.

1.5 Dissertation Organization

The rest of this dissertation is organized as follows. First, Chapter 2 discusses related works. Then, Chapter 3 describes the location prediction framework in more detail. Chapter 4 discusses the opportunistic worker selection using human mobility prediction framework. Finally, in Chapter 5, we conclude this dissertation by summarizing our contributions and discussing future work.

Chapter 2

Background and Related Works

2.1 Human Mobility Analysis

The study of human movement using modern monitoring technologies like mobile phones [34], GPS [35], WiFi [36], and RFID devices [37] has received a lot of attention recently. A variety of research has been conducted with the goal of identifying features of human movement [38–41], for example, divided human mobility into three categories: geographic, temporal, and social connection. In terms of spatial features, they concentrated on geographic mobility, or how far and where a person moves. Temporal factors were taken into account, such as pauses, which represent the amount of time a person spends in a given area. Inter-contact time was defined as the elapsed time between two adjacent contacts for a pair of people, whereas the connectedness characteristic indicates contact or encounter between two people. It is worth noting that incorporating human mobility features aids in precisely predicting human movement. Thus, a mobility prediction model is developed in this dissertation, which takes into account features of human movement such as time, location, and social correlation.

2.2 Location Prediction

Song et al. [45] found significant stability in the predictability of human mobility by calculating the entropy of an individual’s trajectory. Many research attempts have been made thus far to transform this predictability into practical location prediction models [46–48]. The majority of early location prediction approaches are based on patterns [39–41]. The authors of [40] suggested WhereNext, a method for predicting the next position of a moving item with a high degree of precision. The prediction is based on previously extracted movement

Table 2.1: Comparison of some location prediction approaches.

Categories	Strengths	Weaknesses	Paper
Pattern base-models	Models are simple	Models suffer from the one-sided nature of pre-defined patterns	[39–41]
Markov based-models	Models are simple	Models are unable to detect the long-term impact and periodicity of a person’s historical movements.	[11, 42–44]
Deep learning-based models	Models are improved in terms of prediction accuracy	Models are more complicated and require large amount of data	[12–14]

patterns known as trajectory patterns, which are a simple representation of moving object behaviors as sequences of regularly visited places with a standard journey duration. In [41], the authors proposed geographic-temporal-semantic-based location prediction (GTS-LP), a novel mining-based location prediction approach that considers a user’s geography-triggered, temporally triggered, and semantically triggered intentions in order to estimate the likelihood of the user visiting a location. The above methods extract pre-defined mobility patterns (e.g., sequential patterns, periodic patterns) from trajectory traces, and use these patterns to predict future positions. However, these techniques suffer from the one-sided nature of pre-defined patterns.

Several studies attempted to predict where a person will visit based on past knowledge about that individual’s historic locations [11, 42–44]. The Markov model is used in the majority of these investigations. For example, the authors in [11] expanded the Mobility Markov Chain (MMC) model to include n prior visited sites, and they built a unique future location prediction technique based on this mobility model, named n -MMC. According to the authors, the suggested Markov model has higher prediction accuracy than the original Markov model. In [42], a novel division method for pre-processing trajectory data was proposed in which the original trajectory’s feature points are extracted based on structural changes in the trajectory, and then important locations are extracted by clustering the feature points with an improved density, peak-clustering algorithm. Finally, a multi-order fusion Markov model based on the AdaBoost algorithm predicts the next important location of mobile users. These methods, on the other hand, are unable to detect the long-term impact and periodicity of a person’s historical movements.

Deep learning approaches, particularly recurrent neural networks (RNNs) like LSTM [49]

and GRU [50], have recently become popular for capturing long-term sequential impacts and movement patterns. The authors of [12] proposed an RNN-based architecture that learns the embedding of many factors (e.g., user, location, time, keyword) as well as the transition parameters of a recurrent neural network. As a result, it successfully captures semantics-aware spatio-temporal transition regularities to increase accuracy in location prediction. The authors in [13] proposed an attention recurrent network for mobility prediction. With historical mobility attention, the authors suggested a latent variable model that infers the user’s next location. However, training attention networks is challenging in location prediction because they require a large amount of data to achieve optimal parameters. More recently, the authors in [14] predict a vehicle’s likely destinations and routes based on the most recent partial trajectory and contextual data. Nonetheless, their model did not consider the vehicle ID, which aids the model in distinguishing the pattern of vehicles and improves prediction accuracy. A comparison of approaches presented in this subsection can be found in Table 2.1.

The above methods, however, usually do not take mobility information from the POI’s companions into consideration. In fact, incorporating companions’ mobility information might improve model performance when predicting a POI’s next location. In our proposed framework, we first define and select companions of the POI. Then, we train a BRNN model that uses mobility information from the POI and her/his companions to predict the POI’s future locations.

2.3 Social Correlation Detection

Human decision-making is heavily impacted by social contacts [51–53]. Studies on the similarity of human mobility [54] show that social ties are a substantial influencing factor in people’s mobility patterns. This has given rise to the possibility of improving human mobility models by including companion movement. Graph clustering has been used in a lot of research to find social groups in which a network is separated into disjoint communities using clustering techniques [55]. Several studies on community detection have been published based on network members’ contact histories, such as encounter frequency and length, and on a person’s total number of previous contacts [56, 57]. Authors in [56] proposed group discovery using co-location (GDC) and decentralized GDC (DGDC) methods, which use the frequency and length of meetings to reliably discover groups. In [57], the authors used models that can reliably assess, predict, and cluster multi-modal data from people and

groups within a population’s social network to reveal the structure inherent in everyday activities. By evaluating the Bluetooth-based encounter history from a real-life mobility dataset, the work in [58] found social groupings within a network of mobile users. The community discovery approach focuses on creating similarity measurements that can depict the degree of social ties between users by taking into account spatio-temporal components of human interactions, and then using clustering algorithms.

These social correlation detection approaches, however, have drawbacks. They are unable to evaluate many aspects of human movement at the same time, such as location, time, and personal preference. Furthermore, if the data size increases (e.g., a large number of people, a long experimental period), these methods become more time-consuming. Our proposed social correlation detection (i.e., the companion selection method) is based on a deep learning technique that can incorporate many mobility features and that handles big datasets.

2.4 Tasks Allocation in Mobile Crowd Sensing

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 [59]. 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 [60–62]. 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 [63–65]. 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 [66–68]. 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.

2.5 Opportunistic Worker Selection in Mobile Crowd Sensing Using Human Mobility Prediction

To address the challenges associated with OW selection and enhance data collection efficiency, the incorporation of mobility prediction models becomes crucial [69, 70]. 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 [31, 71, 72]. For instance, in iCrowd [31], 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 [73]. 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 [31].

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 [32]. 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.

To address these limitations, in this dissertation, 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.

Chapter 3

Companion Mobility to Assist in Future Human Location Prediction

3.1 Problem Definition

Assume a day is divided into equal time slots (e.g., 15 minutes), and POI i is currently in time slot t . This dissertation considers the problem of predicting the location of POI i in future time slot $t + m$ ($m \geq 1$) given that the recent mobility information of POI i and his/her companions are known. Instead of only predicting the POI's location in the very next time slot (i.e., $m = 1$), the proposed model is designed to predict the POI's locations in several time slots based on different time gaps (e.g., example, $m \in \{1, 2, 3\}$). Terms used in this chapter are listed and defined in 3.1.

Let L_t^i be the location of person i in time slot t , and let X be the recent spatio-temporal information (e.g., location, day of the week, time slot of the day, ID) of POI i and his/her companions. Note that $X = (X_{t-k+1}^i, X_{t-k+2}^i, \dots, X_t^i)$, where k ($k \geq 1$) is the number of recent time slots. The next location prediction is now considered as a classification problem, and the objective of this work is to learn $\hat{P}(L_{t+m}^i|X)$, which is the probability that user i will visit each location during time slot $t + m$.

To illustrate the workings of the prediction model, we've provided a simplified toy example depicted in Fig. 3.1. In this scenario, we assume there are three locations denoted as $L = A, B, C$, a set of recent time slots with $k = 3$, and the POI is associated with two companions. Let's delve into the details of how the model operates.

Suppose we are at a specific time slot t . The next locations prediction model's primary objective is to predict the POI's locations in subsequent time slots, namely $t + 1, t + 2, t + 3$

Table 3.1: Table of term for location prediction framework

Term	Meaning
L_t^i	One-hot vector that indicates the location of person i in time slot t
ID_i	One-hot vector that indicates the ID of person i
C_i^j	One-hot vector that indicates the ID of the j^{th} companion of person i
$L_t^{C_i^j}$	One-hot vector for location of the j^{th} companion of person i in time slot t
D_t	One-hot vector that indicates the day of the week
T_t	One-hot vector that indicates time slot t
E_l	Location embedding matrix
E_{id}	Person ID embedding matrix
k	Number of time slots to look back
POI	Person of interest
PIE	Person ID embedding

(where $m \in 1, 2, 3$ denotes the time slot index). Now, let's focus on the illustration in Fig. 3.1 to understand how the model generates predictions.

For time slot $t + 1$, the model produces an output represented as $(0.1, 0.6, 0.3)$. These values correspond to the likelihood or probability distribution of the POI's presence at each of the three locations: A, B, C . In this particular example, the highest probability is associated with location B , with a probability of 0.6. Therefore, based on this prediction, the model suggests that the POI is most likely to be at location B during time slot $t + 1$. Similarly, the POI is most likely to be at location C during time slot $t + 2$ and $t + 3$.

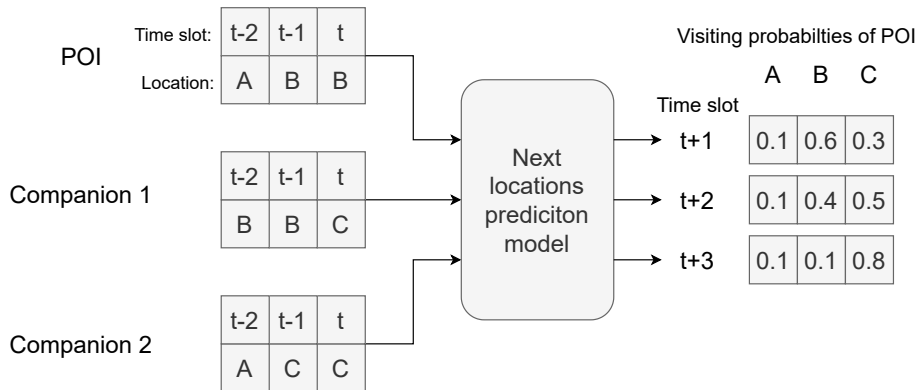


Figure 3.1: A toy example of how the prediction model work with $k = 3, L = \{A, B, C\}, m \in \{1, 2, 3\}$.

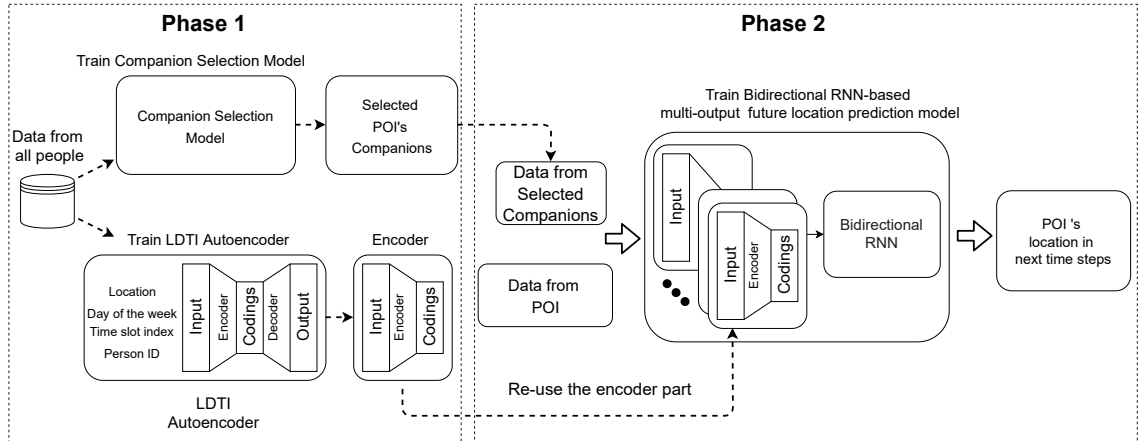


Figure 3.2: Two-phase framework for predicting an individual's future locations.

This simplified toy example serves as a visual representation of how the model generates location predictions for the POI at subsequent time slots, taking into account factors such as past mobility patterns and the influence of companions. It showcases the practical application of the model in predicting future locations based on this multi-dimensional analysis.

3.2 Methodology

In this section, we first provide an overview of the proposed two-phase framework, and we then discuss in detail how the proposed framework selects the POI's companions and predicts her/his future locations.

Figure 3.2 depicts an overview of the proposed method, which consists of two phases. In Phase 1, two models are trained. The first is a selection model to determine the POI's companions. The second model in Phase 1 is the location, day, time, ID autoencoder (LDTI-AE), which is trained to reconstruct a concatenated vector of those features. The encoder in the LDTI-AE is reused as one layer of the prediction model in the second phase.

In Phase 2, the BRNN-based multi-output model is trained. This model takes data from the POI and his/her companions as input and predicts the POI's future location for the next several time slots. In this model, the encoder from the LDTI-AE (e.g., the model trained in Phase 1) is used to compress input features (e.g., location, time slot index, the day of the week, and ID) into dense latent representations that help to mitigate the curse of dimensionality. In the remainder of this section, the details of each part are presented.

3.2.1 Companion Selection Methods

In this subsection, two companion selection methods are presented for identifying the POI's companions. Each uses different metrics to estimate the similarity between mobility patterns. The most similar companions are then chosen. The key advantage of these methods is that they just use the correlation in movement behavior of people, rather than requiring the information on actual social relationships between them. Note that, even if a person moves alone, the companion selection methods still can identify her/his companions.

1. Spatial closeness: The first method uses spatial closeness [15] metric. This metric compares the geographic distributions of individuals to measure the closeness between them in terms of mobility.
 - (a) Location-person vector construction: First, a location-person vector with a length equal to the number of locations is constructed for each person. Each element in the vector presents the probability of an individual being seen at a specific location. Table 3.2 shows samples of location-person vectors.

Table 3.2: Samples of location-person vectors.

Person	Location 1	Location 2	...	Location N
u	0.064	0.453	...	0.088
v	0.082	0.621	...	0.001
...

- (b) Spatial closeness calculation: The SC score between the two people with vectors u and v is calculated as shown below:

$$\text{SC score} = 1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2} \quad (3.1)$$

The SC score indicates the distance between vectors u and v in the location-person vector space. Those who usually visit the same location (i.e., companions) will get a low SC score, otherwise, the SC score between vectors will be high.

2. Person ID embedding model: The second companion selection model leverages the concept of embedding, a powerful technique extensively utilized in neural networks, particularly in natural language processing and recommendation systems. Embedding enables the transformation of discrete variables into continuous vectors of real-valued numbers, which are more amenable to processing by neural network architectures.

This process involves mapping each discrete variable to a low-dimensional vector in a continuous space.

In the context of the proposed framework, embedding is applied to the discrete features related to mobility patterns, such as location, day of the week, and time slot. By representing these categorical variables as continuous vectors, the model can better capture the underlying semantic relationships and contextual nuances within the data. This means that similar features are grouped together in the embedding space, allowing the neural network to grasp the intricate dependencies and patterns that may not be apparent when dealing with raw categorical data.

The advantage of employing embedding in the second companion selection model is that it enables a more sophisticated understanding of mobility patterns. It allows the model to identify not only straightforward correlations but also subtle and complex relationships between the POI and potential companions based on their movement behavior. Consequently, the framework can make more informed and context-aware decisions when selecting companions for predicting the POI's future locations. In the following sections, we will delve into the technical specifics of this embedding technique and its implications for the framework's predictive accuracy.

(a) Person-ID-embedding matrix:

An embedding-based companion selection model is developed to find a group of companions whose mobility information can help to predict the POI's future locations. The core idea behind this model involves leveraging the power of embeddings, specifically Person ID Embeddings (PIE), to represent individuals and capture their mobility patterns. Instead of dealing with discrete person IDs, the model transforms them into continuous PIE vectors that encapsulate the unique characteristics and behaviors of each individual. This transformation process enables a more nuanced understanding of the people's mobility, which, in turn, is used to predict POI movements.

The model's functionality relies on cosine similarity, a widely adopted metric in similarity measurement, to gauge the resemblance between the PIE vector of a given POI and the PIE vectors of other individuals within the dataset. This similarity score essentially serves as a gauge for assessing the compatibility and relevance of individuals as potential companions for the POI in the context of

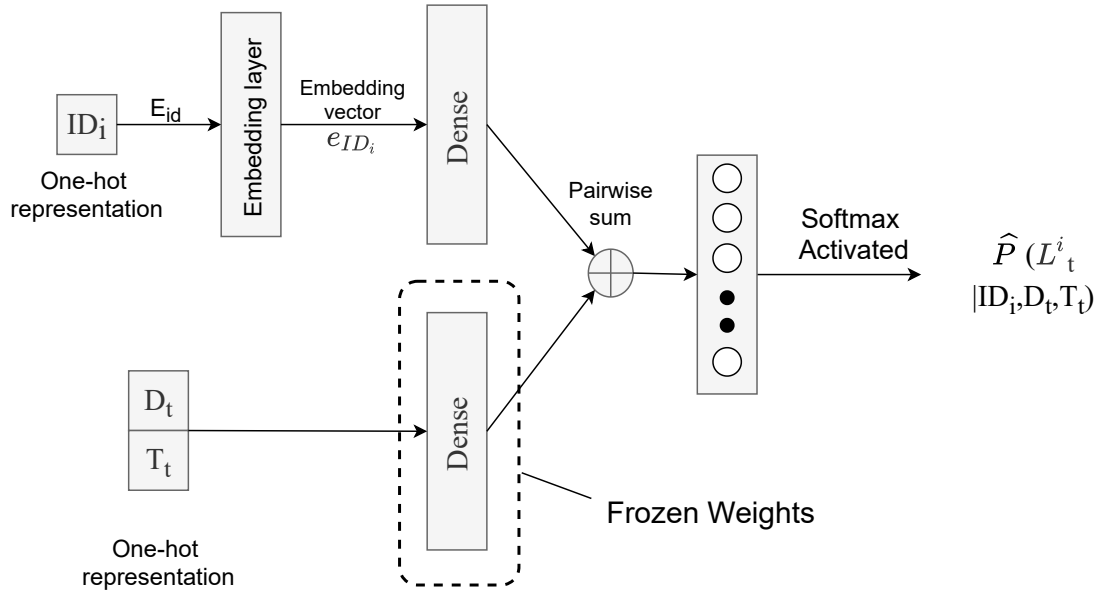


Figure 3.3: Model architecture for learning a PIE matrix in phase 1.

predicting future locations.

The architecture to train the PIE matrix is shown in Fig. 3.3. The model is characterized by two input branches, each responsible for handling distinct facets of information.

In the first branch, the model takes in the person’s unique identifier (ID_i), typically represented as a one-hot vector. This identifier is then subjected to a transformation process involving the PIE matrix (E_{id}), which ingeniously converts the discrete person ID into a PIE vector (e_{ID}). This PIE vector is the crux of the model’s ability to understand and encapsulate the mobility characteristics of each individual, essentially transforming it into a continuous and informative representation.

The second branch of the model receives input in the form of one-hot vectors representing the day of the week (D_t) and the time slot index (T_t). These inputs are crucial in understanding the temporal aspects of mobility, as they provide context regarding when an individual’s mobility patterns are expected to manifest.

Then, each of the two branches is fed into a dense layer, as shown in Fig. 3.3, and those two dense layers have the same number of neurons (e.g., the number of neurons equals the number of locations). Please note that the weights of the dense layer in the second branch are intentionally frozen. This decision is driven by the understanding that the temporal aspects (day of the week and time slot)

are relatively stable and should not be altered during the training process, in contrast to the dynamic person ID embeddings.

Lastly, outputs from the two dense layers are pairwise summed and softmax activated to estimate the individual's current location (L_t^i).

- (b) Similarity measurement: Given the PIE matrix, a cosine similarity metric is used to calculate the similarity score between PIE vectors u and v of two people, as shown below:

$$\text{Cosine similarity score} = \frac{u \cdot v}{\|u\|_2 \|v\|_2} \quad (3.2)$$

As described above, temporal information (e.g., the day of the week, the time slot index) and spatial information (e.g., locations) are used to train a model that can estimate a person's current location. Note that weights of the dense layer in the second branch are frozen (as shown in Fig. 3.3), and therefore, the model only updates parameters of the dense layer and the embedding layer in the first branch. This requires the PIE matrix to be updated in a way that effectively reflects the correlation between a person and his/her visited locations. So, if two people usually move in a similar way (i.e., visit the same place at a specific time), their PIE vectors are forced to be close to each other in the embedding space. Thus, the similarity score between these two PIE vectors will be high. On the other hand, if the two people's movements differ, their PIE vectors in the embedding space are far apart, and their PIE vectors' similarity score is low.

3.2.2 Location-Day-Time slot-ID Autoencoder (LDTI-AE)

The implementation of autoencoders has been fundamental in the realm of artificial neural networks, enabling the acquisition of condensed representations known as latent representations or codings, which offer a high degree of utility in various applications. Notably, autoencoders excel in dimensionality reduction by transforming complex input data into codings of substantially lower dimensionality. In our research, we introduce a novel concept – the Stacked Location, Day, Time, and ID Autoencoder (LDTI-AE), specially designed to process one-hot vectors representing location, day of the week, time slot, and person's ID.

The LDTI-AE functions as a reconstruction engine, taking the one-hot vectors as input and effectively deciphering them back to their original form. This process results in the LDTI-AE generating dense latent representations that encapsulate the intricate dynamics

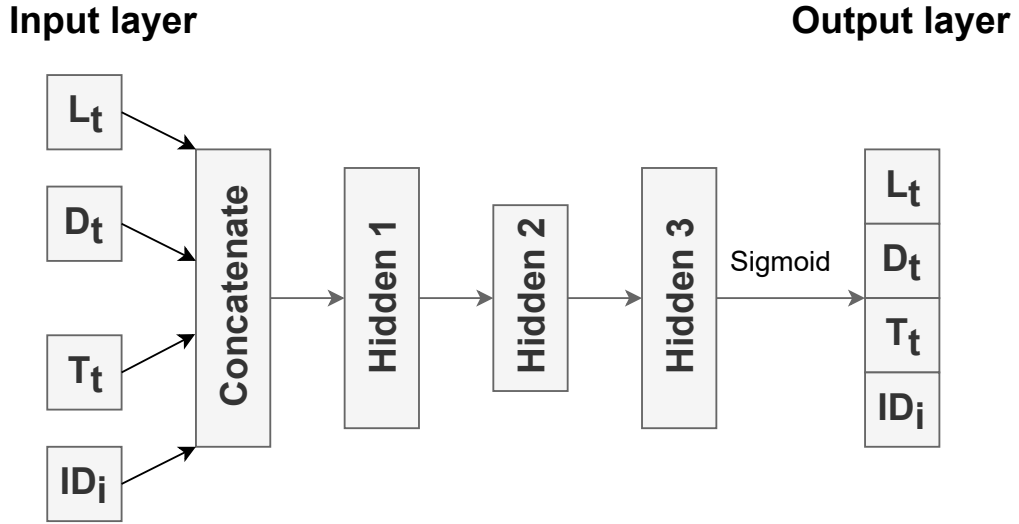


Figure 3.4: LDTI - Autoencoder architecture in phase 1.

of user mobility. Our proposed LDTI-AE architecture, as illustrated in Fig. 3.4, consists of three hidden layers in addition to the input and output layers. It operates on the one-hot vectors, namely L_t for location, D_t for the day of the week, T_t for the time slot index, and ID_i for the person’s ID.

To empower the learning process within the hidden layers, we employ the Rectified Linear Unit (ReLU) activation function for the first two hidden layers. Given the concatenated nature of the output vector, comprising L_t , D_t , T_t , and ID_i , we leverage the sigmoid activation function within the output layer. Furthermore, the training process is steered by a cross-entropy loss function, optimizing the LDTI-AE to effectively reproduce the original one-hot vectors.

This LDTI-AE design is an important contribution of our framework, demonstrating its capability to distill spatio-temporal and personal attributes into compact latent representations, offering a critical dimensionality reduction that enhances the overall efficiency of our predictive models. Its significance lies in its potential to discern intricate patterns within mobility data, a pivotal feature in enabling the effective prediction of a person’s future locations.

3.2.3 BRNN-based Future Location Prediction Model

To predict future locations, the proposed model adopts a Bidirectional Recurrent Neural Network (BRNN) configuration, featuring the utilization of Long Short-Term Memory (LSTM) cells. This innovative model excels in deciphering the complex dynamics of human mobility,

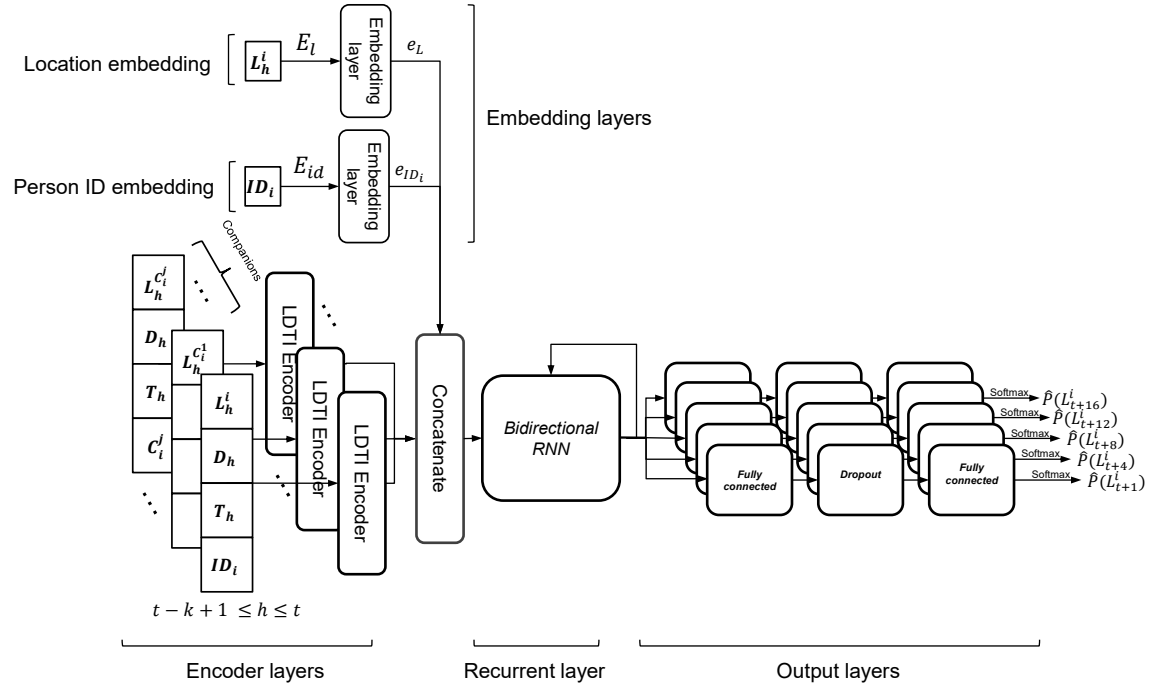


Figure 3.5: The BRNN-based future locations prediction model.

making it an integral component of our framework.

Leveraging data from the POI and two selected companions, the BRNN-based multi-output model undertakes the task of predicting the locations that the POI may potentially visit within subsequent time slots. Specifically, these time slots include $t + m$, where m spans a selection of intervals such as 1, 4, 8, 12, and, 16. The architecture of this model is thoughtfully illustrated in Fig. 3.5, serving as a visual representation of its inner workings.

One of the noteworthy components of this model is the adoption of Bidirectional Recurrent Neural Networks. This architectural choice empowers the model to analyze sequences in both forward and backward directions, enhancing its ability to capture intricate temporal dependencies within the data. Furthermore, the incorporation of LSTM cells enables the model to effectively manage and remember crucial information over extended sequences. This dynamic memory retention plays a pivotal role in ensuring the model's proficiency in understanding and predicting the POI's future locations.

The introduction of this BRNN-based multi-output model represents a significant stride in the development of our predictive framework. Its role is to translate intricate mobility patterns into actionable predictions, thereby enhancing the framework's overall predictive accuracy and enabling a more profound understanding of human mobility dynamics. The

subsequent sections will delve deeper into the individual elements and the model’s overall functionality, elucidating its significance within our framework.

As shown in Fig. 3.5, the proposed model consists of four layers: the encoder layer, the embedding layer, the recurrent layer, and the output layer. Details of each layer are as follows.

1. The encoder layers: These employ the encoder (i.e., up to hidden layer 2 of the LDITI-AE) to compress the spatio-temporal and personal features of the POI and his/her companions into dense latent representations. Note that all encoder weights are frozen in this phase.
2. The embedding layers:
 - (a) Location embedding: This layer learns embedding matrix, E_l , such that every location L can be transformed into embedding vector e_L .
 - (b) Person ID embedding: The ID embedding layer learns embedding matrix, E_{id} , such that the ID of person i , ID_i , can be transformed into embedding vector e_{ID} . Please note that embedding matrix E_{id} in this layer is different from the PIE matrix that is described in the Companion Selection Model subsection.
3. The recurrent layer: Output from the encoder and embedding layers is then concatenated before being fed into the recurrent layer. In this work, the recurrent layer is a BRNN that uses LSTM cells.
4. The output layers: The last hidden state of the LSTM cell then passes through five identical branches, each of which includes a fully connected layer with ReLU activation, a dropout layer, and a fully connected layer with a softmax activation function. As illustrated in Fig. 3.5, each branch corresponds to predicting the POI’s location in a certain time slot. In particular, the output of the branch that predicts the location of POI i in time slot $t + m$, $m \in \{1, 4, 8, 12, 16\}$ is $\hat{P}(L_{t+m}^i)$, which is the probability the POI will visit each location in time slot $t + m$.

3.2.4 Parameter Learning

To train and update model parameters, the cross-entropy loss function is utilized. The total loss of the five outputs is computed as follows:

$$\text{Total loss} = \sum_{\text{batchsize}} \sum_m -\log(\hat{P}(L_{t+m})) \quad (3.3)$$

where $m \in \{1, 4, 8, 12, 16\}$ corresponds to output of the prediction model; $\hat{P}(L_{t+m}^i)$ is the model's predicted probability for the POI's location in time slot $t + m$.

In practical applications, the utility of location predictions tends to decrease as we look further into the future. Recognizing this, our model incorporates a weighted loss mechanism to account for the varying importance of predictions across different time slots.

This weighted loss approach assigns a higher penalty to the loss term associated with predictions for the near-future time slots, reflecting the higher significance of these predictions. Conversely, it assigns a lower penalty to the loss term for predictions pertaining to distant-future time slots, which are less critical.

To calculate the weighted total loss of the five output loss terms, the following procedure is followed:

$$\text{Weighted total loss} = \sum_{\text{batchsize}} \sum_m -W_m \log(\hat{P}(L_{t+m}^i)) \quad (3.4)$$

where

$$W_m = \frac{1}{m} \times h; h = \frac{1}{\sum_{n \in \{1,4,8,12,16\}} \frac{1}{n}} = \frac{48}{73} \quad (3.5)$$

is the weight for the loss term of the output that predicts the POI's location in time slot $t + m$; h is the constant that keep $\sum_{m \in \{1,4,8,12,16\}} W_m = 1$

3.3 Datasets

In our study, we employed two extensive datasets of Wi-Fi traces for training and evaluating our proposed framework. Given the limited coverage range of Wi-Fi technology, the location of the connected access point (AP) can be regarded as the mobile user's location at that particular moment [74, 75]. Consequently, we represent human mobility as a sequence of connected APs. The first dataset utilized is the Dartmouth dataset [17], which encompasses Wi-Fi logs from 13,888 mobile device users on the Dartmouth College campus. These logs include a timestamp, device ID, and the basic service set identifier (BSSID) recorded each

Table 3.3: Statistics on the two experimental datasets after pre-processing.

	Dartmouth	UB
Number of people (mobile users)	50	50
Number of locations (APs)	623 + 1 dummy	1243 + 1 dummy
Number of time slots per day	41	41
Experiment period	118 days	90 days

time a mobile device associates or disassociates with a Wi-Fi AP. The second dataset, known as the Buffalo/phonelab-wifi dataset (UB dataset) [18], encompasses five months of data originating from smartphones carried by 284 members of the University at Buffalo (UB), including faculty, staff, and students.

For the training and testing of our models, we selected data from the Dartmouth dataset pertaining to the 50 most active mobile users over a period of 118 days, spanning from January 3 to April 30, 2004. In this experiment, we focused on 623 access points (locations). Similarly, from the UB dataset, we selected data from the 50 most active mobile users over 90 days, in conjunction with 1243 access points.

As a crucial part of our data preparation, we addressed variations in data collection intervals between the two datasets. We segmented a typical working day, running from 8:00 to 18:00, into time slots. Subsequently, we mapped the timestamp of each data record to its corresponding pre-defined time slot. Notably, mobile users tend to remain at a particular location for a significant duration, such as during a class. In this study, we standardized the time slot duration to 15 minutes. Consequently, there were a total of 41 time slots per day, including the final time slot. It's important to clarify that, within a specific time slot, the user's most recent location is considered their location for the entire time slot. Additionally, time slots devoid of data records were tagged with a designated placeholder location.

To provide an overview of the dataset statistics following this pre-processing stage, please refer to Table 3.3.

Due to the sparsity of data in both datasets, a significant proportion of time slots was labeled with a placeholder, often referred to as a "dummy location." This led to a challenging issue where the model, due to the imbalance in data, frequently predicted this dummy location, which held no practical value. To mitigate this problem, a data processing step was applied. Specifically, all training and validation samples labeled with the dummy location were removed. In other words, all five labels for the five distinct outputs corresponding to time slots $t + 1$, $t + 4$, $t + 8$, $t + 12$, and $t + 16$ were required to have real locations instead of

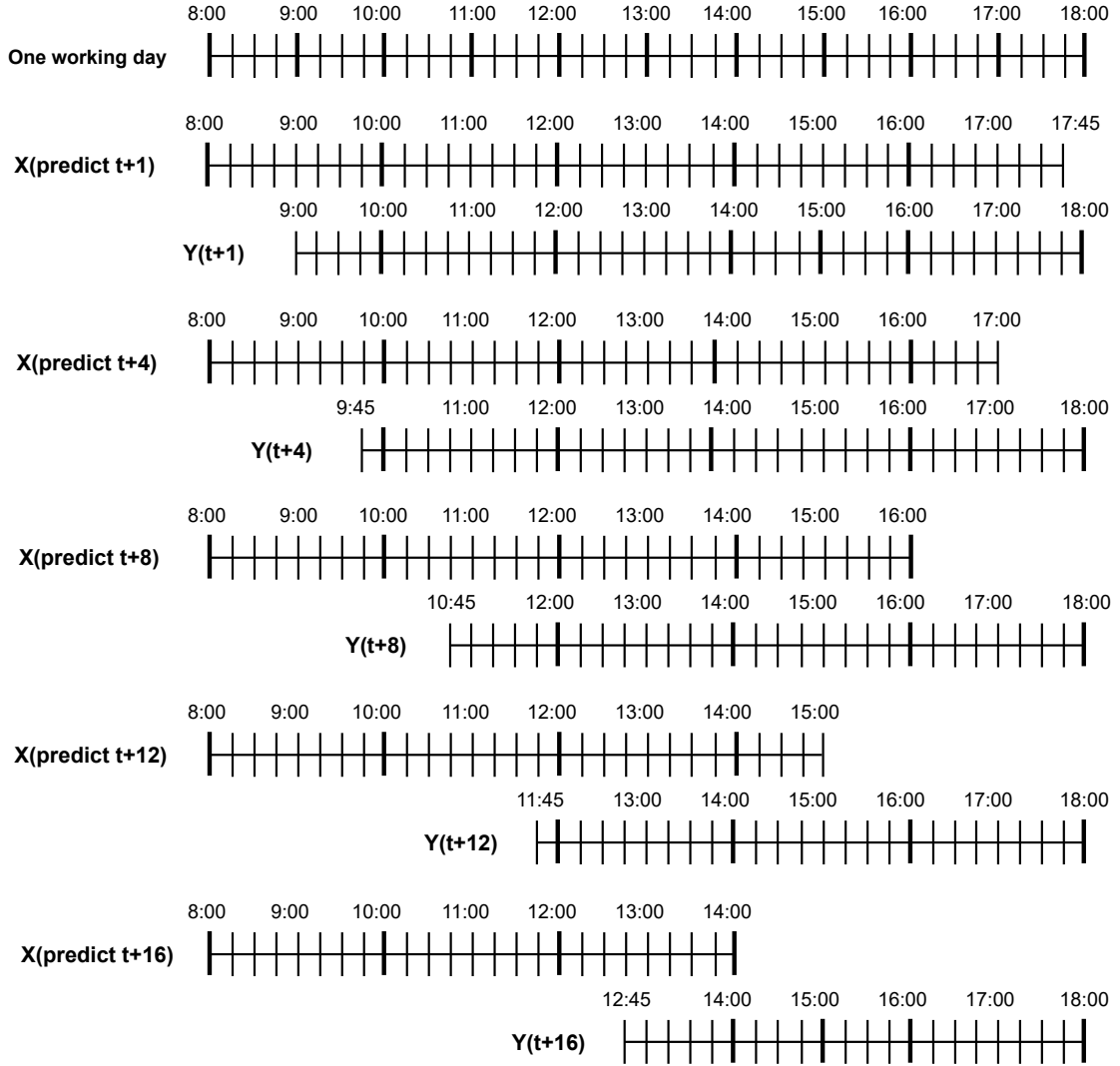


Figure 3.6: Input-Label timelines of five test sets.

the dummy location.

It's important to clarify that this trimming of dummy locations pertained exclusively to the labels, while dummy locations could still exist in the input features. Additionally, to ensure that the model had the capacity to predict locations for all the time slots between 9:00 and 18:00, additional time slot data ranging from 8:00 to 22:00 was incorporated into the training phase.

For the testing phase, a fair and independent assessment of the model was conducted for each distinct output. During this phase, only the label corresponding to the specific output being evaluated was required to have a real location (non-dummy location), while labels for the other outputs could still contain dummy locations. To elucidate, when assessing the model's performance in predicting the location for time slot $t + 1$, it was only necessary

to ensure that the label for output $t + 1$ was not the dummy location. Consequently, five distinct test sets were created for the testing phase to evaluate the model’s performance for each of the five outputs: $t + 1, t + 4, t + 8, t + 12$, and $t + 16$. A visual representation of these input-label timelines can be seen in Figure 3.6.

To provide further context, it’s worth noting that in both the UB and Dartmouth traces, the experiment periods spanned 90 and 118 days, respectively. For the temporal dimension, each dataset was partitioned randomly into training, validation, and test sets following a 6:2:2 ratio. The first two sets were designated for model training and hyperparameter optimization, while the final set was reserved for the evaluation of model performance. This division scheme ensured a balanced and comprehensive assessment of the model’s capabilities.

3.4 Experiment Results

In this section, we provide the results from a series of experiments carried out to evaluate the efficacy of our proposed framework. We begin with the baseline models and then move on to the implementation details. Finally, we present the experiment results followed by validation of the proposed approaches.

3.4.1 Baseline Models

Our proposed model underwent a comparative analysis against several baseline methods, each offering distinct approaches to the task of location prediction. These baseline methods included:

- Markov Model [11]: This method focuses on learning and establishing movement regularities in human mobility patterns. It employs these regularities to select the location with the highest probability for a future visit.
- SERM [12]: SERM is designed to capture the diverse contextual factors influencing human movement. It achieves this by learning embedding vectors for multiple variables, including ID, location, time, and keywords. Additionally, SERM models the transition parameters of a recurrent neural network to enable semantics-aware predictions of the next location.
- VANext [13]: VANext is centered around a variational attention mechanism that combines past mobility pattern periodicity with recent check-in preferences. This fusion

of historical patterns and recent preferences enhances the accuracy of next-location predictions.

- DPBPT [14]: DPBPT operates by predicting the next location based on the most recent locations.

These baseline methods, each with its unique strategy and underlying principles, served as benchmarks against which our proposed model was evaluated. This comparative analysis enabled us to gauge the performance and effectiveness of our model in the context of location prediction, shedding light on its strengths and potential advancements in this field.

3.4.2 Implementation Details

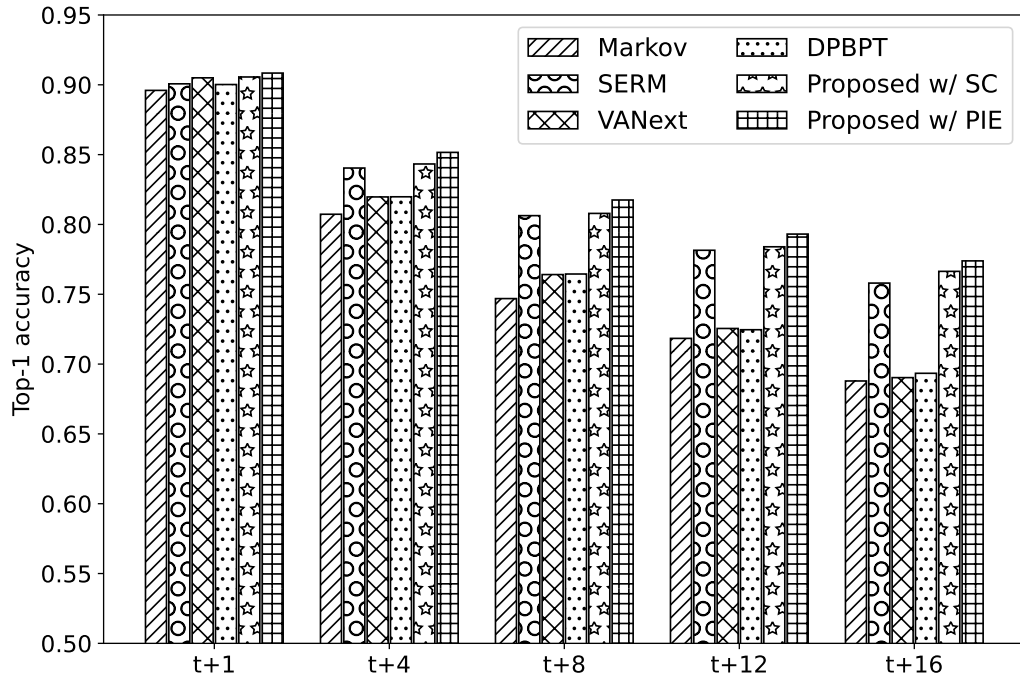
The implementation of both the proposed and baseline models was carried out using the TensorFlow Keras library. The hardware setup for this implementation involved a four-core Xeon CPU paired with a single Titan-XP GPU, ensuring computational efficiency and performance.

During the training process, each model underwent 50 training epochs to ensure convergence and optimize performance. To identify the most effective set of hyperparameters, we relied on validation accuracy as a key metric for model selection.

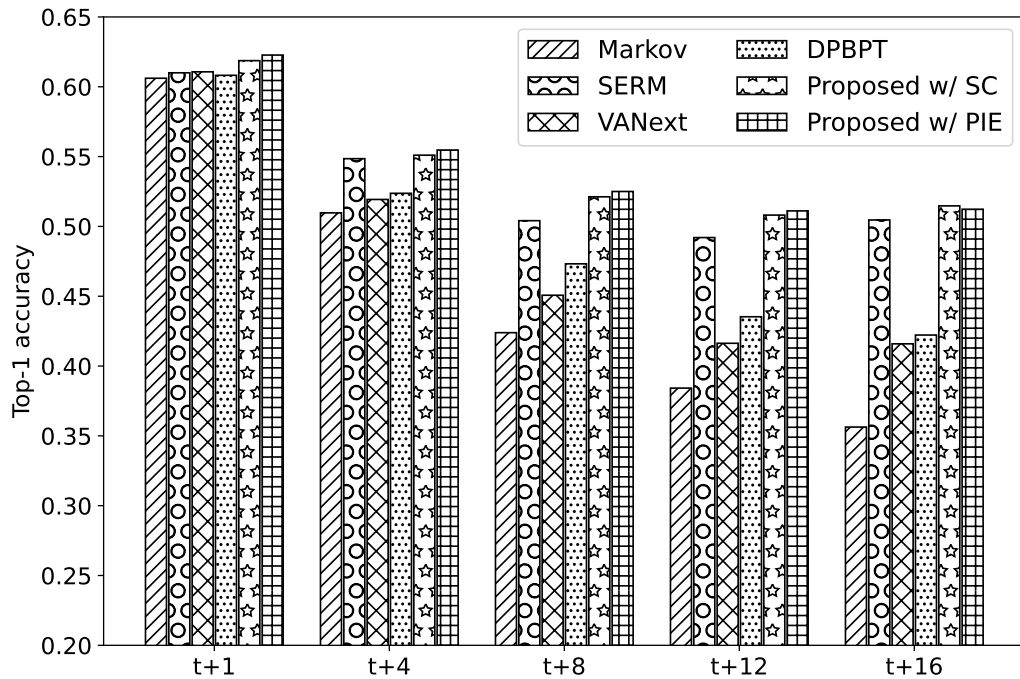
The optimization of the models was accomplished through the RMSprop optimizer, with a learning rate parameter set at 0.001. Furthermore, we configured a batch size of 1024 for the training process. This batch size choice helps to balance the computational load and memory usage during training.

For the baseline models, a dedicated tuning process was conducted to fine-tune their parameters, ensuring the best performance for each specific dataset. This meticulous tuning procedure was performed to adapt the baseline models to the unique characteristics and complexities of the datasets under consideration, ultimately maximizing their predictive capabilities.

In summary, our implementation setup, hyperparameter selection, and optimization strategies were carefully tailored to harness the computational resources efficiently and to yield the best possible performance from both the proposed model and the baseline methods. This methodical approach allowed for a fair and insightful comparative evaluation of the models in the context of location prediction.



(a) Dartmouth dataset.



(b) UB dataset.

Figure 3.7: Top-1 accuracies on the two datasets from the different methods predicting POI locations in several time slots.

3.4.3 Experiment Results

Figure 3.7 shows the top-1 test accuracy from the proposed and the baseline models when predicting an individual’s locations in the next time slots (e.g., $t+1$, $t+4$, $t+8$, $t+12$, and $t+16$). Comparisons of these methods resulted in the following.

RNN-based models (SERM, VANext, the proposed model with SC, and the proposed model with PIE) outperformed the Markov-based model by a large margin when predicting the target person’s location in future time slots, owing to two factors. First, the Markov model is based on assumptions about personal movement distributions. Second, Markov models can only model first-order dependencies for a person’s movement characteristics, while RNN-based models can model long-term dependencies. Note that the performance of the Markov model when predicting the individual’s location in the next time slot, $t+1$, was high thanks to the characteristics of the user’s movements in the two datasets, where mobile users frequently stayed in one place for a couple of time slots.

The performance of the VANext model and the DPBPT when predicting an individual’s location in distant time slots dropped significantly. The major issue is that both models do not consider the person ID, which has an important role in future location prediction. The person ID helps the model distinguish mobility characteristics between people and makes for a more accurate prediction when estimating that person’s future location. SERM was a strong baseline when predicting a person’s location in future time slots cause this model considered the person’s ID.

Models that utilized companion mobility information (e.g., the suggested model with SC and PIE) outperformed models that did not use companion mobility information (Markov, SERM, VANext, and DPBPT). This demonstrates that using companion mobility information improves the model’s accuracy. Models using the PIE approach surpassed the models that use the SC approach because the PIE approach incorporates both spatial and temporal aspects in people’s movements, whereas the SC technique just considers the spatial component.

The average top-1, top-3, and top-5 test accuracies of the five outputs of different models are shown in Table 3.4 and Table 3.5. As can be seen, the proposed models that consider companion mobility (with SC and with PIE) outperformed baseline models that do not include companion information. Again, this proves that human movement is highly dependent on companions; consequently, incorporating companion mobility information into the

Table 3.4: Average top-1, top-3, and top-5 accuracies with Dartmouth data.

Model	Avg. Top-1 Accuracy	Avg. Top-3 Accuracy	Avg. Top-5 Accuracy
Markov	0.7713	0.8703	0.8964
SERM	0.8174	0.9376	0.9573
VANext	0.7809	0.8865	0.9013
DPBPT	0.7805	0.8878	0.9052
Proposed model with SC	0.8214	0.9431	0.9582
Proposed model with PIE	0.8289	0.9462	0.9590

Table 3.5: Average top-1, top-3, and top-5 accuracies with UB data.

Model	Avg. Top-1 Accuracy	Avg. Top-3 Accuracy	Avg. Top-5 Accuracy
Markov	0.4560	0.6496	0.7089
SERM	0.5318	0.7401	0.7970
VANext	0.4824	0.6523	0.7126
DPBPT	0.4925	0.6648	0.7210
Proposed model with SC	0.5427	0.7466	0.8043
Proposed model with PIE	0.5452	0.7499	0.8095

prediction model enhances performance when predicting the POI’s future locations.

Table 3.6 shows the number of parameters for each model on the two experimental datasets. In general, the proposed models (with SC and PIE) have fewer parameters than the SERM and DPBPT models, demonstrating the effectiveness of our proposed model. Although the VANext model has the fewest parameters, its performance in distant time slots is significantly lower than other models.

Table 3.6: Number of parameters of different models on the two datasets.

Model	Dartmouth	UB
SERM	442,572	607,336
VANext	168,451	341,140
DPBPT	586,672	1,026,252
Proposed model with SC	354,096	554,576
Proposed model with PIE	414,144	521,788

3.4.4 Sensitivity of Hyper-parameters

The impact of different model hyper-parameter settings on location prediction performance is investigated in this subsection. Specifically, we evaluate the sensitivity of four hyper-parameters: person ID embedding size, the location embedding size, the number of bi-LSTM hidden units, and the first fully connected layer size. We set the person ID embedding size and the location embedding size to $\{8, 16, 32, 64, 128\}$, the number of bi-LSTM hidden units

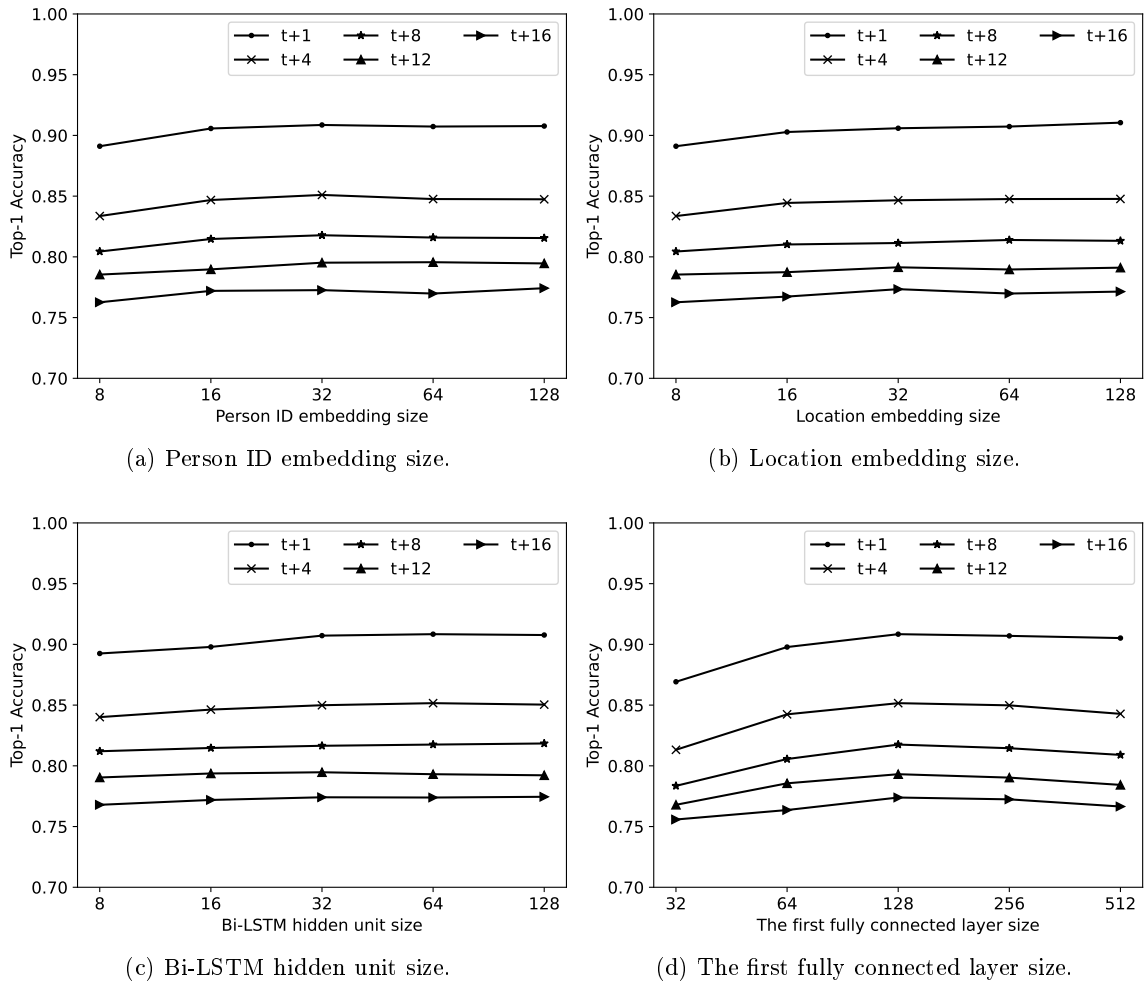


Figure 3.8: Performance with varying parameters on Dartmouth dataset. (a) Person ID embedding size. (b) Location embedding size. (c) Bi-LSTM hidden unit size. (d) The first fully connected layer size.

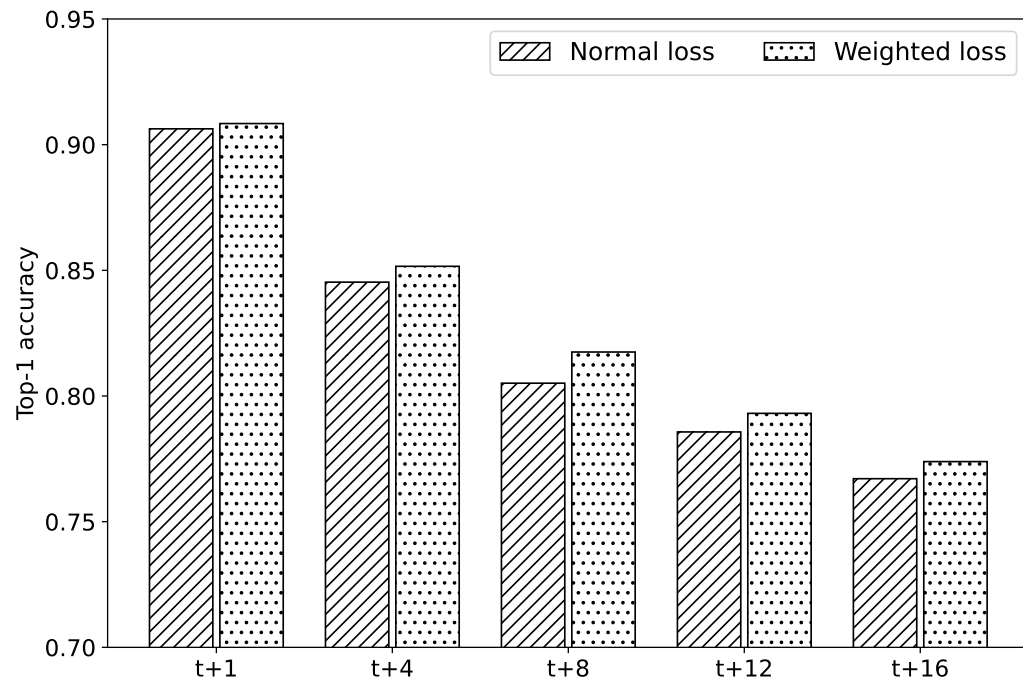
to $\{8, 16, 32, 64, 128\}$, and the first fully connected layer size to $\{32, 64, 128, 256, 512\}$. Except for the parameters under test, all other parameters were left at their default settings. Figure 3.8 shows the performance comparison when we vary the values of model parameters on the Dartmouth dataset. In general, we find that using a larger value of the hyper-parameters improves performance by allowing for a more powerful representation. However, this increases the model’s complexity and makes it prone to overfitting. In our experiments, based on the results in Fig. 3.8, we chose the following hyper-parameters to account for the effectiveness vs. computational cost trade-off: the location and user embedding sizes are set to 64, the number of bi-LSTM hidden units is set to 64, and the size of the first fully connected layer is set to 128.

3.4.5 Effectiveness of Weighted Loss

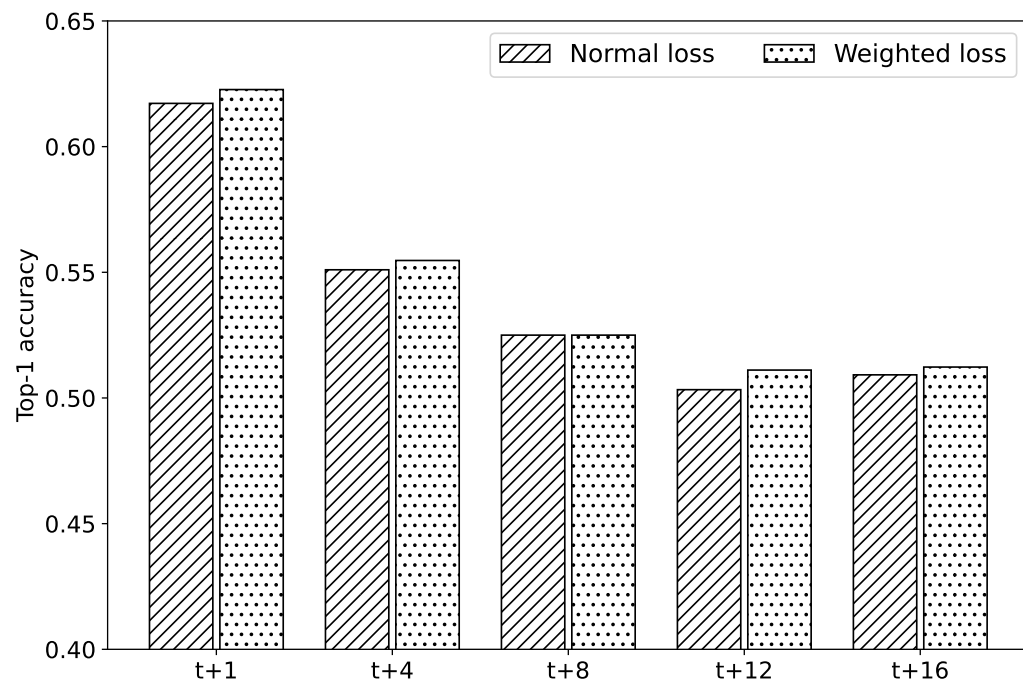
In this subsection, we conduct an in-depth examination of the effectiveness of using weighted loss in enhancing the performance of our proposed model. Figure 3.9 provides a visual representation of how the model, employing PIE as a key training element, performs on our two experimental datasets when utilizing both normal loss and weighted loss.

The graph in Figure 3.9 vividly illustrates the tangible benefits of weighted loss in elevating the accuracy of the model, particularly when predicting the POI’s location in near-future time slots, such as $t + 1$ and $t + 4$. As previously mentioned, the incorporation of weighted loss essentially compels the model to prioritize the precision of predictions for these imminent time slots. Consequently, this focused optimization leads to a noticeable enhancement in the model’s performance within this temporal context.

However, it is worth noting that the advantages of weighted loss are not limited to the prediction of locations in the near future. When evaluating the model’s performance in predicting locations in more distant-future time slots, like $t + 8$, $t + 12$, and $t + 16$, we observed notable improvements across both experimental datasets. The rationale behind this phenomenon may be attributed to the unique characteristic of weighted loss. By assigning different weights to the loss term associated with each output of the model, it inherently influences the learning rate while fine-tuning the model parameters with respect to the outputs. In essence, this dynamic and adaptive learning rate mechanism plays a pivotal role in consistently enhancing the accuracy of a multi-output model, iteratively refining its predictions with each subsequent output.



(a) Dartmouth dataset



(b) UB dataset

Figure 3.9: Performance comparison of the proposed model with PIE, trained using normal loss and weighted loss.

3.4.6 Person ID Embedding Companion Selection Validation

In this subsection, the proposed PIE-based companion selection method is validated. This step's purpose is to demonstrate that the PIE vectors of two people with similar mobility are close to each other in the embedding space. Otherwise, their PIE vectors remain apart. For the demonstration, the t-SNE technique [76] was employed to visualize the PIE vector.

The Dartmouth dataset was used for this subsection. Besides the original data from 50 people, synthetic data of four new virtual people (IDs 51, 52, 53, and 54) were generated from the original data. The data preparation is described as follows.

- Step 1: Pick training data of five people at random from a pool of 50 original people. Here, randomly chosen people are the persons with IDs 1, 34, 44, 10, and 50.
- Step 2: Create data of the five new virtual people by randomly combining data samples of the five people chosen in Step 1. The synthetic data construction of the five new virtual people (IDs 51, 52, 53, and 54) is shown in Fig. 3.10. Please note that the data of each chosen person were randomly selected at the sample level. For example, to create data for virtual person 51, 50% of the sample data of person 1, and 50% of the sample data of person 34 were randomly chosen and combined.

Data of person 51	50% of data from person 1	50% of data from person 34
Data of person 52	50% of data from person 1	50% of data from person 44
Data of person 53	25% of data from person 1	75% of data from person 10
Data of person 54	100% of data from person 50	

Figure 3.10: Synthetic data construction for five new virtual people with IDs 51, 52, 53, and 54.

The cosine similarity scores of the PIE vectors for specific individuals are presented in Table 3.7. Analyzing these scores leads to several insightful conclusions as below.

- Firstly, it is worth noting that the source data used to construct virtual persons' data

Table 3.7: The cosine similarity scores between different people’s PIE vectors.

Person ID	1	34	44	10	50	51	52	53	54
1	1.0000	0.1832	0.1705	0.1364	0.0976	0.6910	0.7467	0.5697	0.0982
34	0.1832	1.0000	0.1378	0.2357	0.1576	0.8014	0.2076	0.2044	0.1975
44	0.1705	0.1378	1.0000	0.0618	0.2942	0.1378	0.7351	0.0618	0.2942
10	0.1364	0.2357	0.0618	1.0000	-0.2157	0.2357	0.1630	0.8559	-0.2157
50	0.0976	0.1576	0.2942	-0.1860	1.0000	0.1576	0.3149	-0.1860	0.9999
51	0.6910	0.8014	0.1378	0.2357	0.1576	1.0000	0.5259	0.5150	0.1570
52	0.7467	0.2076	0.7351	0.1630	0.3149	0.5259	1.0000	0.4234	0.3158
53	0.5697	0.2044	0.0618	0.8559	-0.1860	0.5150	0.4234	1.0000	-0.1893
54	0.0982	0.1975	0.2942	-0.2157	0.9999	0.1570	0.3158	-0.1893	1.0000

(e.g., persons 1, 34, 44, 10, and 50) was chosen randomly. Notably, their person-id-embedding (PIE) vectors exhibited relatively low similarity scores, as illustrated in Table 3.7. Specifically, the cosine similarities between the PIE vector of person 1 and persons 34, 44, 10, and 50 were 0.1832, 0.1705, 0.1364, and 0.0976, respectively. This suggests that the data for virtual users 51 and 52 originated from source data positioned far apart in the embedding space. However, the cosine similarity scores between the PIE of person 51 and 52 with their source data person are relatively high. This indicates that even when information is synthesized from disparate sources, the resulting node maintains a high similarity with its source nodes.

- Comparatively, the cosine similarity score between the PIE vectors of person 53 and person 1 is lower than those of person 51 and person 1, and person 52 and person 1 (e.g., 0.5697 vs. 0.6910 and 0.7467, respectively). This discrepancy is attributed to person 53’s mobility data utilizing only 25% of the data from person 1, while persons 51 and 52 utilize 50%.
- With 75% overlap in mobility data from person 10, the cosine similarity score between the PIE vectors of persons 53 and 10 is notably high (e.g., 0.8559).
- Persons 51 and 52 share the same 50% of mobility data from person 1, resulting in relatively high cosine similarity scores among them. For example, the cosine similarity score between the PIE of persons 51 and 52 is 0.5259.
- The cosine similarity score between the PIE vectors of individuals 50 and 54 is nearly perfect, at approximately 1 (e.g., 0.9999), indicating identical mobility patterns. Similarly, high similarity scores are observed between PIE vectors of 44 and 52, 10 and 53, 51 and 34, highlighting similarities in their respective mobility profiles.

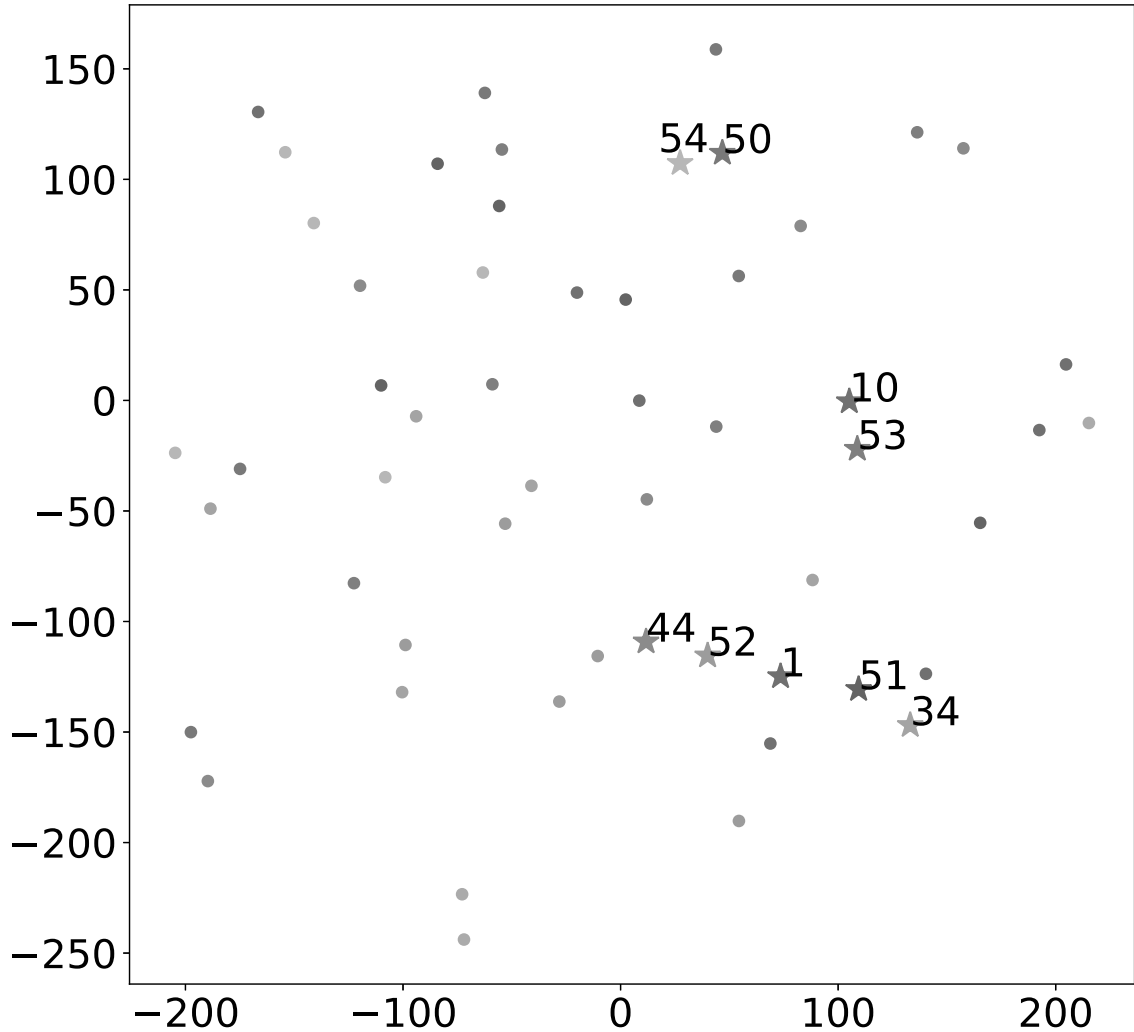


Figure 3.11: t-SNE plot of the PIE vectors for all 54 people. Each point in the plot indicates a PIE vector.

Given the data from 54 people (e.g., 50 original and 4 virtual), a PIE model was trained to get the PIE vectors, which were then visualized into two-dimensional space using the t-SNE technique. Figure 3.11 shows the t-SNE plot of the PIE vectors for all 54 people. Each point in the plot indicates a PIE vector. We focused on the PIE vectors of the following IDs: 1, 10, 34, 44, 50, 51, 52, 53, and 54.

Some observation can be made from the t-SNE plot:

- The mobility patterns of both person 50 and virtual person 54 are identical, resulting in closely positioned PIE vectors for this pair. However, it's noteworthy that although the PIE vectors for individuals 50 and 54 exhibit high similarity, the corresponding t-SNE algorithm [76] projections may not perfectly overlap due to inherent randomness

in the algorithm.

- A similar pattern is observed for other pairs, such as 44 and 52, 10 and 53, 51 and 34.
- The PIE vectors of individuals in group 51, 52, and 53 closely cluster around the PIE vector of person 1, indicating substantial similarity in their mobility profiles compared to person 1.
- We acknowledge that the t-SNE plot may not fully capture dissimilarity between node (e.g., person id embedding (PIE) vectors) for the individuals under consideration. For instance, as shown in Table 3.7, the similarity score between person 1 and person 34 is low (0.1832), but their positions in the t-SNE plot (as shown in Fig. 3.11) appear relatively close.

The main reason is that during the dimensionality reduction process, t-SNE emphasizes selecting pairs with high similarity and compressing them to remain in close proximity in low-dimensional space. Consequently, pairs such as 1-51 and 34-51 appear close, leading to the placement of pair 1-34 in proximity, despite their low similarity score (0.1832).

This tendency is also reflected in the distribution of nodes 1, 34, 44, 10 in Fig. 3.11. For example, nodes 1 and 44 exhibit low similarity scores, yet t-SNE pulls node 1 and node 44 close to node 52, creating the appearance of close proximity in the t-SNE figure.

The utilization of the t-SNE plot in this manuscript serves the purpose of illustrating that nodes with high similarity scores share similar positions in the embedding space. However, since the above mentioned limitation of the t-SNE method – nodes with low similarity scores appear close in the 2D plot, as seen with nodes 1, 34, 44, and 10.

3.5 Chapter Summary

In this chapter, we present a comprehensive two-phase framework designed to predict a person’s future locations with a strong focus on enhancing predictive accuracy. Our innovative framework leverages the rich spatio-temporal contexts embedded within the mobility patterns of both the person of interest (POI) and their companions. By intelligently selecting companions whose mobility information contributes significantly to predicting the POI’s future locations, our framework introduces two efficient methods. The first method hinges

on spatial proximity, while the second takes advantage of a novel embedding technique. Additionally, we address the challenge of high dimensionality by integrating a stacked autoencoder, which learns a latent vector capturing essential factors governing human mobility, including location, day of the week, time slot, and person ID. To culminate the predictive process, we employ a bidirectional recurrent neural network (BRNN)-based multi-output model, utilizing mobility information from the POI and their companions while ensuring robust training through weighted loss. This holistic approach enables accurate predictions of the POI’s locations across several future time slots.

It’s noteworthy that our framework does not rely on social relationships between the POI and companions but instead on the similarity in their movement behavior. This distinction makes our model highly versatile, and applicable to a wide range of spatio-temporal datasets, even in cases where social relationship information is lacking.

To validate the efficacy of our model, we conducted rigorous evaluations on two real-world datasets. The results unequivocally indicate that our framework surpasses baseline methodologies in predicting a person’s future locations. These findings emphasize the significant potential of our framework in enhancing the accuracy and efficiency of human mobility prediction, positioning it as a promising approach for a broad spectrum of applications in various domains.

Chapter 4

Context-Aware Worker Recruitment for MobileCrowd Sensing Based on Mobility Prediction

4.1 System Model and Problem Definition

4.1.1 System Model

This study centers on the challenge of Mobile Crowd Sensing (MCS), a domain wherein a multitude of sensing devices is strategically deployed across urban landscapes to amass data concerning a diverse array of phenomena. These phenomena span a spectrum of variables, including but not limited to traffic conditions, air quality, and noise levels. At the inception of each day, a comprehensive set of sensing tasks is strategically distributed across the urban landscape, presenting a unique logistical puzzle for the MCS data center. This challenge requires the MCS data center to accomplish as many of these designated tasks as possible, all within the constraints of a predefined budget.

To achieve this goal, the MCS center has two primary strategies at its disposal: it can opt to engage participatory workers, such as mobile users or taxi drivers, who actively contribute to data collection, driven by attractive incentives; alternatively, the center can seek the assistance of Opportunistic Workers (OWs), who require comparatively lower incentives for their contributions.

In the context of this research, our primary focus is on the intricate task of strategically selecting OWs with the objective of optimizing task completion within the constraints of a

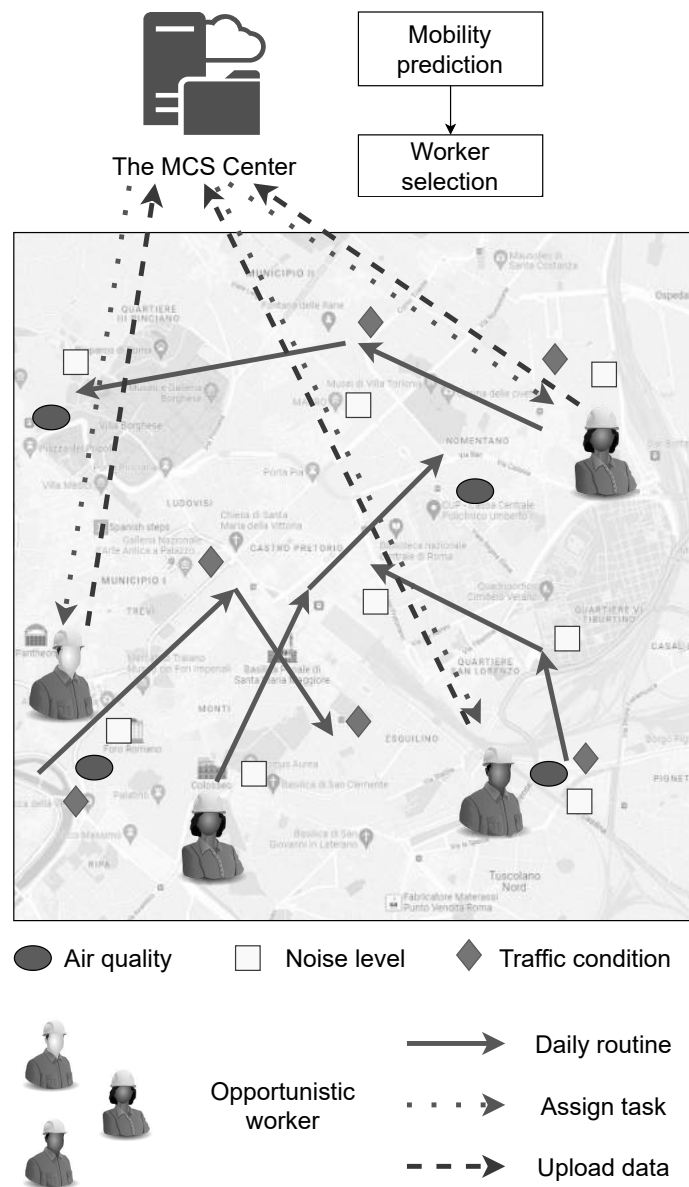


Figure 4.1: The system model for this study.

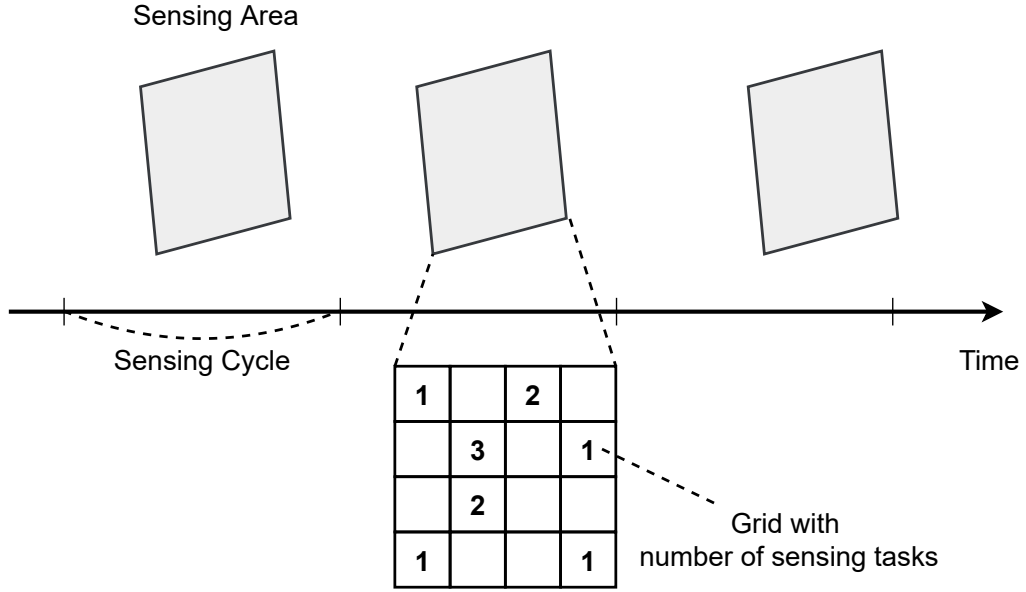


Figure 4.2: Example of a task map.

fixed budget. The intricate web of these activities is captured and illustrated in Figure 4.1, which delineates the data collection framework involving the recruited workers, illuminating the intricate dynamics of this complex system.

4.1.2 Problem Definition

The sensing area is partitioned into k smaller sections called grids, which are represented as $G = \{g_1, g_2, \dots, g_k\}$. The working day is divided into m SCs, which are represented as $T = \{t_1, t_2, \dots, t_m\}$. At the start of each day, the MCS organizer assigns a number of tasks to each grid in every SC. Figure 4.2 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 $W = \{w_1, w_2, \dots, w_n\}$ 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:

$$\text{Maximize } |C|$$

Table 4.1: Table of term for OW selection framework

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

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

Terms used in this work are listed and defined in Table 4.1.

Comparison to iCrowd [31] and DLMV [32] 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.

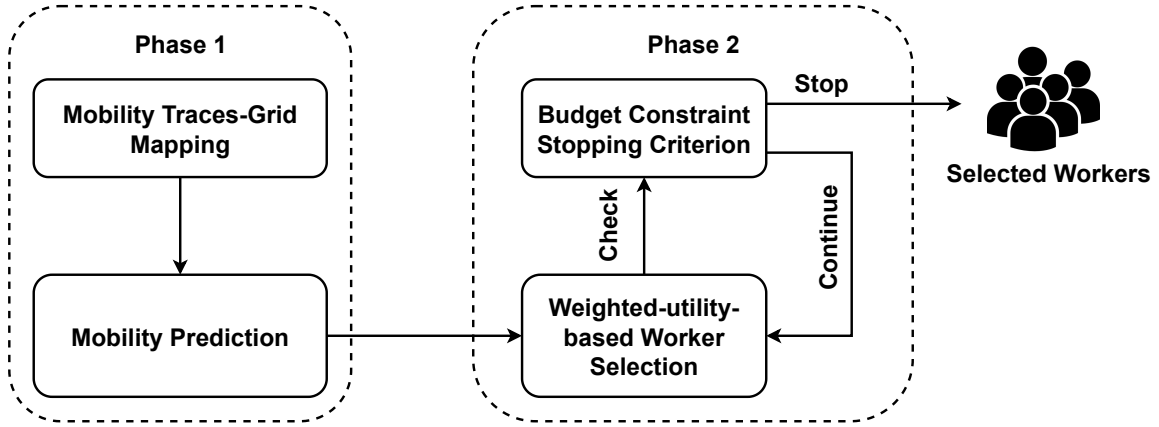


Figure 4.3: The proposed CAMP framework.

4.2 Methodology

4.2.1 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, to predict each worker’s location; and 2) using 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. 4.3 and works as follows.

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

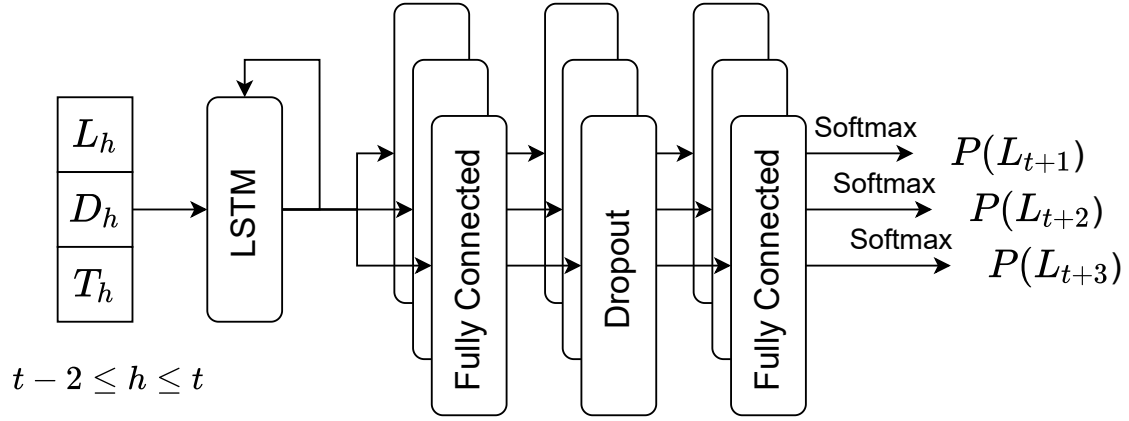


Figure 4.4: The RNN-based multi-output next location prediction model.

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.

4.2.2 Multi-output Next-location Prediction

The recurrent neural network, a type of neural network designed to process sequential data with cycles and internal memory units, is a core component of the CAMP method for predicting workers' future locations. In this context, the CAMP method employs an RNN-based multi-output model with an LSTM (Long Short-Term Memory) cell to predict a worker's potential locations over the next three sensing cycles ($q = 3$). The worker's mobility data serve as input to the model, which, in turn, forecasts the worker's potential locations in the subsequent time slots ($t + k, k \in 1, 2, 3$). The architectural layout of the location prediction model is graphically represented in Figure 4.4.

As depicted in Fig. 4.4, 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. 4.4, 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 is composed of three parallel branches, each including a fully connected layer equipped with a Rectified Linear Unit (ReLU) activation function. This is followed by a dropout layer, which helps prevent overfitting, and another fully connected layer that employs softmax activation. Figure 4.4 illustrates the mapping of each branch, each of which is responsible for predicting the worker's locations within a specific time interval. For instance, the branch associated with predicting the worker's locations during time slot $t + k$, where $k \in 1, 2, 3$, is denoted as $P(L_{t+k})$. This branch estimates the probability of visiting different locations during that specific 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:

$$\text{Loss} = \sum^{\text{batch_size}} \sum_{i=1}^q -\log(P(L_{t+i})) \quad (4.1)$$

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 $t + i$.

4.2.3 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:

$$\begin{aligned} \text{weighted-utility}(W_s) &= \sum_{g \in G} \sum_{i=1}^q \text{weight}(g) \\ &\times \text{prob}_{g,t+i}(W_s) \end{aligned} \quad (4.2)$$

where:

- W_s is the set of selected workers.
- $\text{weight}(g)$ 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.(4.3).

$$\text{prob}_{g,t+i}(W_s) = 1 - \prod_{w \in W_s} (1 - P_{g,t+i}(w)) \quad (4.3)$$

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: In this step, we merge each unselected worker, denoted as w (i.e., $w \in W \setminus W_s$), with the existing set W_s to create a new set ($W_s \cup w$). This is accomplished by considering the available information, specifically, W , W_s , and the task map for the current SC. Subsequently, the weighted-utility of each combined set is computed as $\text{weighted-utility}(W_s \cup w)$ using Eq. (4.2). The combined set displaying the highest weighted-utility is designated as the new W_s for the forthcoming iteration. This process iterates until the total incentives allocated to selected workers surpass the budget constraint. Algorithm 1 provides the pseudo-code detailing the worker selection process for reference.

Algorithm 1 Weighted-utility Opportunistic Worker Selection.

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

- 1: **set** $W_s = \emptyset$
- 2: **while** $|W_s| \times I \leq B$ and $W \neq \emptyset$ **do**
- 3: **set** MaxUtility = 0, bestW = 0
- 4: **for** $w_j \in (W \setminus W_s)$ **do**
- 5: Calculate weighted-utility($W_s \cup w_j$) using Eq.(4.2)
- 6: **if** weighted-utility($W_s \cup w_j$) > MaxUtility **then**
- 7: MaxUtility = weighted-utility($W_s \cup w_j$)
- 8: bestW = w_j
- 9: **end if**
- 10: **end for**
- 11: $W_s = W_s \cup \text{bestW}$
- 12: $W = W \setminus \text{bestW}$
- 13: **end while**
- 14: **return** W_s

4.3 Experimental Results

4.3.1 Baseline Methods

In our comprehensive evaluation, we rigorously compared the performance of the CAMP framework with two alternative methodologies: the iCrowd framework [31] and the DLMV framework [32]. These frameworks represent distinct strategies for addressing the intricate challenge of worker selection within MCS.

The iCrowd framework operates on the premise of an inhomogeneous Poisson process for mobility predictions. This approach utilizes a greedy maximum utility algorithm to make informed decisions regarding worker selection. In contrast, the DLMV framework adopts a deep learning-based approach to predict vehicle mobility, leveraging advanced neural networks to enhance mobility forecasting. Task assignment within the DLMV framework is facilitated through a greedy online algorithm, aiming to optimize worker-task pairings.

By pitting the CAMP framework against these well-established methodologies, we sought to provide a comprehensive assessment of its effectiveness in the crucial task of selecting OWs in MCS. This comparative analysis allows us to discern the relative strengths and weaknesses of each approach, shedding light on the unique contributions and advantages of the CAMP framework in the context of worker selection strategies for MCS.

4.3.2 Dataset and Experiment Setups

The validation of the proposed CAMP framework was carried out utilizing the Crowdad Roma/Taxi dataset, a well-established and reliable source of real-world data. This dataset comprises the GPS coordinates of approximately 320 taxis, meticulously collected over a 30-day period spanning from February 1, 2014, to March 3, 2014. Each taxi’s trajectory is characterized by a series of GPS points, each accompanied by precise timestamps.

To ensure the robustness and credibility of our evaluation, extensive data-cleaning processes were employed to refine the dataset. Following this data refinement, a total of 314 taxis were selected to serve as the workers for our MCS scenario. This comprehensive dataset was instrumental in simulating real-world conditions and ensuring the relevance of our evaluation.

In our validation procedure, the dataset was partitioned to facilitate the training of the prediction model and the subsequent evaluation of the OW selection algorithm. The first 25 days’ worth of data were meticulously harnessed to train and fine-tune the prediction model, leveraging the rich historical mobility data. The remaining 5 days’ data was designated for the evaluation phase, during which the performance of the OW selection algorithm was rigorously assessed under operational conditions. This systematic approach to dataset utilization mirrors real-world operational scenarios, thus underlining the practical applicability of the CAMP framework in diverse MCS applications.

For the experimental phase, a $5km \times 5km$ square area within Rome’s urban region was thoughtfully selected as the testing ground. To ensure systematic assessment, this chosen area was further subdivided into a grid layout, with each grid measuring $500m \times 500m$. This grid configuration, as visually depicted in Figure 4.5, facilitated a structured experiment setup.

The experiment timeline aligned with a typical working day, spanning from 8:00 in the morning to 18:00 in the evening. This time frame was meticulously divided into 60 Sensing Cycles (SCs), with each SC encompassing a 10-minute duration. This setup allowed for precise time management and observation of worker behaviors throughout the day.

In our experimental design, we introduced variability in the number of tasks generated per SC, ranging from a minimum of 5 tasks to a maximum of 25 tasks. This broad range enabled us to assess the performance of the CAMP framework under diverse task loads, mirroring practical scenarios.

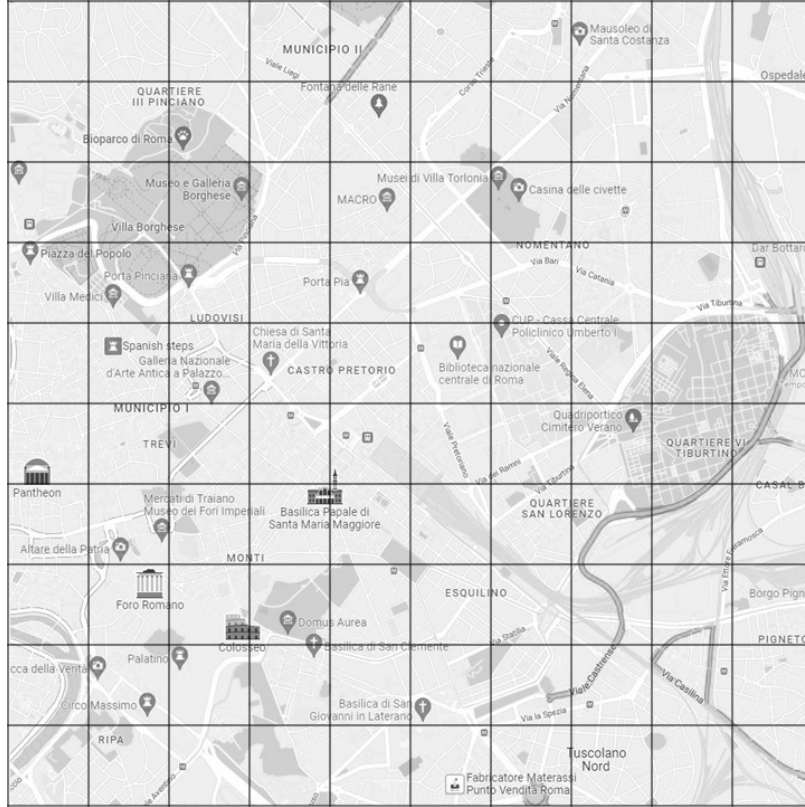


Figure 4.5: Grids map of Rome city.

Furthermore, the budget (denoted as B) for the experiment was systematically configured to take on values of 5, 10, 15, 20, and, 25 while maintaining a constant incentive rate (I) of 1. The selection of these budget values aimed to explore the implications of different budget constraints on worker selection efficiency.

The comprehensive experiment setups are succinctly summarized in Table 4.2, outlining the key parameters and configurations employed to evaluate the performance of the CAMP framework under varying conditions, facilitating a thorough understanding of its capabilities and adaptability in practical Mobile Crowd Sensing (MCS) applications.

Table 4.2: 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

At the beginning of each day, the MCS center initiates the task allocation process by randomly assigning a specific number of tasks across a set of grids, denoted as $|G|$. This allocation procedure is integral in forming the task map that delineates the distribution of tasks across different grids during each Sensing Cycle (SC). This task map is methodically created and allocated at the beginning of each operational day by the MCS center, serving as a foundational blueprint for task distribution and worker recruitment.

In this study, we consider two distinctive scenarios to encapsulate diverse operational settings and applications within the MCS framework:

- Scenario 1: In this scenario, the MCS center maintains a consistent and unchanging task map for all SCs throughout the day. This approach is ideally suited for applications where task requirements remain relatively stable and exhibit minimal fluctuations. For instance, consider a scenario in which tasks involve monitoring air quality levels in a city. In such cases, the demand for tasks within each grid is expected to remain constant over time, allowing the MCS center to utilize a fixed task map that is applicable to all SCs.
- Scenario 2: In contrast, Scenario 2 embodies a dynamic operational mode in which the MCS center modifies the task map for every single SC. This approach is better aligned with applications characterized by dynamic and evolving task demands that vary throughout the day. For example, when tasked with monitoring traffic conditions within a city, the number of tasks required in each grid may fluctuate based on real-time traffic patterns and demands. Consequently, a scenario where the task map is altered at each SC becomes a more relevant and practical approach to effectively cater to changing task requirements.

The consideration of these two scenarios provides a practical and insightful perspective on the nuanced management of task maps within various Mobile Crowd Sensing (MCS) applications. The ability to tailor the task map strategy to match the specific characteristics of the tasks and their temporal variations offers a pathway to optimizing resource allocation and ultimately enhancing the overall performance of the system. This adaptability ensures that the MCS center can effectively respond to the dynamic task requirements inherent in diverse applications, leading to more efficient data collection and improved task completion rates.

For the execution of our experiments, a robust computational setup was employed, fea-

turing a four-core Intel Xeon W-2123 CPU, 32 GB of memory, and a Titan-XP GPU. The implementation of the CAMP framework and the comparison baseline methods was achieved using the TensorFlow Keras library within a 64-bit Python 3.8 environment. This robust computational infrastructure was instrumental in conducting thorough and precise assessments of the proposed framework’s performance and efficiency, ensuring the reliability and relevance of our findings in the context of real-world Mobile Crowd Sensing applications.

4.3.3 Location Prediction Performance

Table 4.3: 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 [32]	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 [32]	26.22	15.02	10.16	17.13
	The statistic model in iCrowd [31]	-	-	-	3.6

In this subsection, we embark on a comprehensive evaluation of the performance of the proposed location prediction model, juxtaposed against existing approaches present in the literature. The performance assessment is carried out by subjecting both the RNN-based multi-output location prediction model introduced in this study and the prediction model featured in the DLMV framework to a rigorous evaluation process, encompassing both validation and test datasets. It is worth noting that the statistics-based models employed in iCrowd underwent evaluation solely on the test dataset, utilizing the identification of the location with the highest probability of being visited as the worker’s next destination.

Table 4.3 unveils the prediction accuracy metrics of the models when forecasting worker locations within subsequent SC – specifically, SC $t + 1$, SC $t + 2$, and SC $t + 3$. The results underscore the efficacy of RNN-based prediction models, notably the CAMP and the DLMV model, outperforming statistics-based models. Significantly, these RNN-based models exhibit the highest accuracy when predicting worker locations in the immediate SC, with a marginal decrease in accuracy as the prediction horizon extends into the future. This

observation signifies the predominant influence of a worker’s recent movement history on their near-term locations, highlighting the instrumental role of recent mobility patterns in forecasting future locations.

These outcomes serve as compelling evidence for the utility of RNN-based models in enhancing worker location predictions within MCS applications. They underscore the pivotal role of considering a worker’s recent mobility patterns as a key determinant in making accurate predictions about their future locations, thus establishing the superiority of RNN-based models in the context of MCS.

4.3.4 Experiment Results

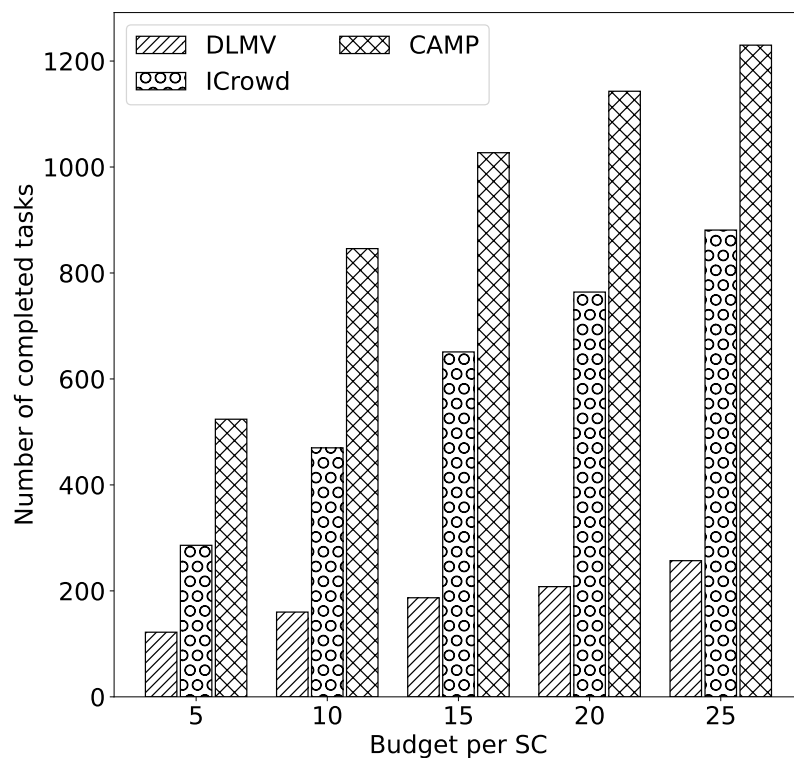
4.3.4.1 Different Budget Values:

This section provides an in-depth analysis of the performance of the Context-Aware Worker Recruitment based on a Mobility Prediction Model (CAMP) framework within various budget constraints and scenarios. We delve into the intricate relationship between the budget allocation for each sensing cycle and the resultant number of successfully completed sensing tasks. Figure 4.6 serves as a visual aid, offering a comparative perspective on performance variations across a range of budget constraints. The budget for each Sensing Cycle (SC) spans from 5 to 25, with the number of tasks generated per SC maintained at the default value of 25.

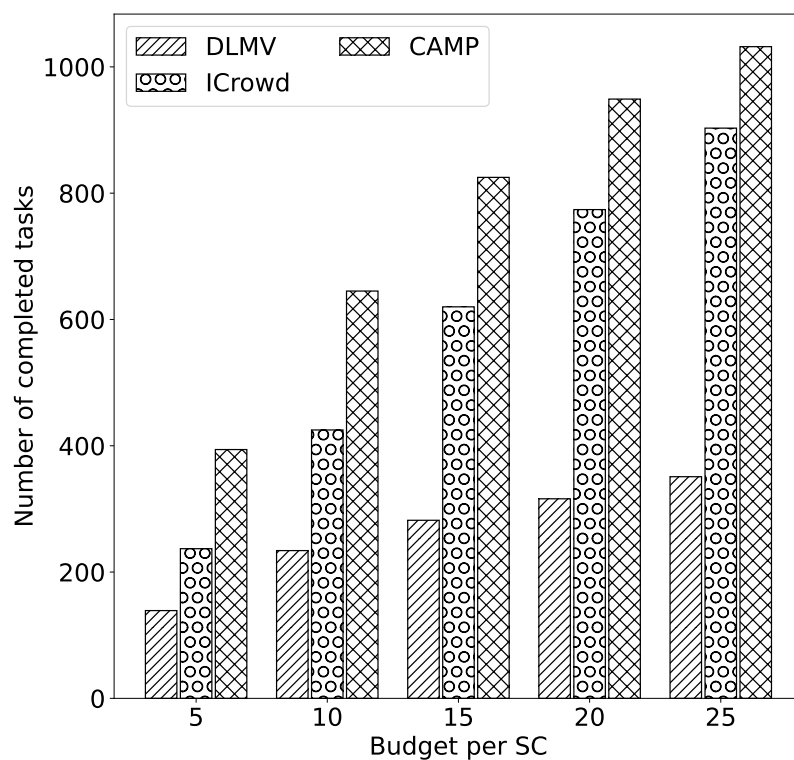
To provide a clearer understanding of the calculations, it’s essential to highlight the following key points:

- 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, we observe a general increase in the number of completed tasks, which can be attributed to the recruitment of additional workers to fulfill the tasks. Notably, CAMP consistently outperforms both ICrowd and DLMV across all budget constraints in terms of the number of completed tasks.



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



(b) Scenario 2: Different task map every SC

Figure 4.6: 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.

In Scenario 1, CAMP achieves 56.3 percent more completed tasks than ICrowd, signifying a substantial enhancement in task completion rates. Similarly, within the same scenario, CAMP surpasses DLMV by 410.7 percent, underscoring the remarkable impact of our proposed framework.

Furthermore, in Scenario 2, CAMP reaffirms its superiority by achieving an average of 29.9 percent more completed tasks than ICrowd. Likewise, within this scenario, CAMP outperforms DLMV by 190.8 percent, reiterating the consistent effectiveness of the CAMP approach in dynamic task environments. These results underscore the significant potential and adaptability of CAMP in enhancing worker selection efficiency across diverse MCS applications.

A notable and compelling finding of our study is the consistent outperformance of the CAMP framework when compared to the iCrowd method. This superiority can be attributed to two fundamental factors. Firstly, the iCrowd method relies on the assumption that workers' mobility adheres to a specific statistical process. However, human mobility patterns are inherently complex and diverse, and this assumption may not accurately capture the intricacies of real-world mobility. In contrast, the CAMP framework leverages a cutting-edge RNN-based multi-output prediction model. This model excels in accurately predicting workers' future locations by drawing insights from their past mobility data. The precision in mobility prediction significantly enhances worker selection, leading to a noticeable performance advantage.

Secondly, a crucial distinction between the two approaches lies in their worker selection strategies. The iCrowd framework overlooks the number of tasks associated with different locations when making worker selections. In contrast, the CAMP framework introduces a weighted-utility worker selection algorithm that strategically assigns higher weights to workers more likely to visit locations with a higher task demand. This approach results in an optimized worker selection process, effectively distributing tasks, and ultimately culminating in higher task completion rates.

Our findings also indicate that while the prediction accuracy of the DLMV and CAMP methods is comparable (as shown in Table 4.3), the number of completed tasks using the DLMV method significantly lags behind. This discrepancy can be attributed to the DLMV approach's reliance on a worker selection algorithm that predominantly considers the likelihood of workers visiting a specific location. Conversely, both the CAMP framework and the iCrowd method adopt a more comprehensive approach by considering a worker's likelihood

of visiting all locations when calculating utility. This holistic evaluation of utility leads to superior performance and a more robust task completion rate.

In summary, the technical analysis of our results underscores the comprehensive strengths of the CAMP framework in effectively predicting worker mobility, optimizing worker selection based on task distribution, and ultimately elevating task completion rates within the dynamic landscape of MCS scenarios.

4.3.4.2 Different Numbers of Tasks

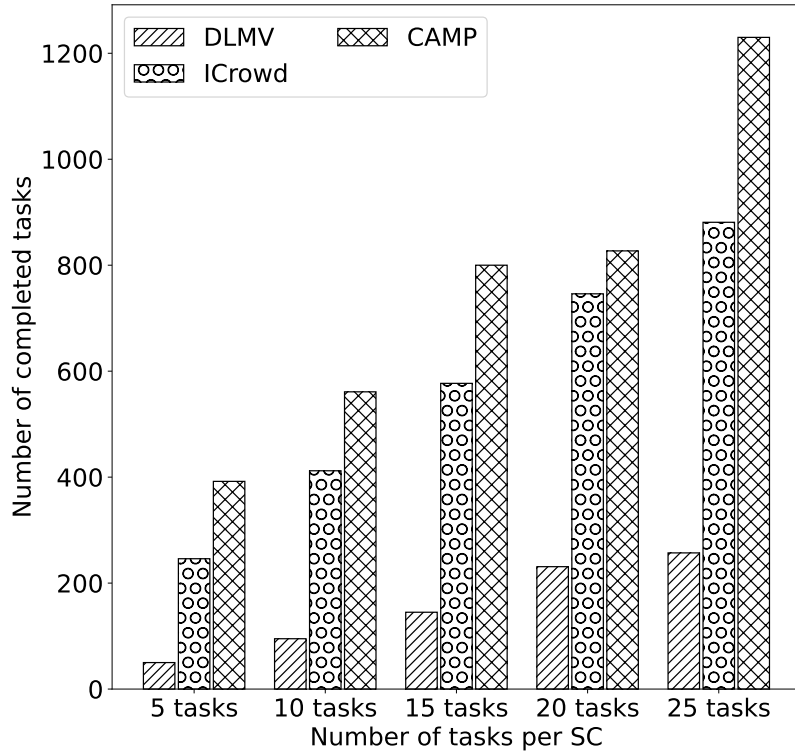
In this section, we extend our performance evaluation of the CAMP framework, delving into an in-depth analysis and providing valuable insights into its remarkable superiority over baseline methods in diverse task load scenarios. Our evaluation encompasses a range of task settings, varying from 5 tasks per sensing cycle (SC) to 25 tasks per SC, offering a comprehensive view of the framework’s capabilities. The budget allocated for each SC in this evaluation remains constant at 25.

The following calculations clarify the experiment setup:

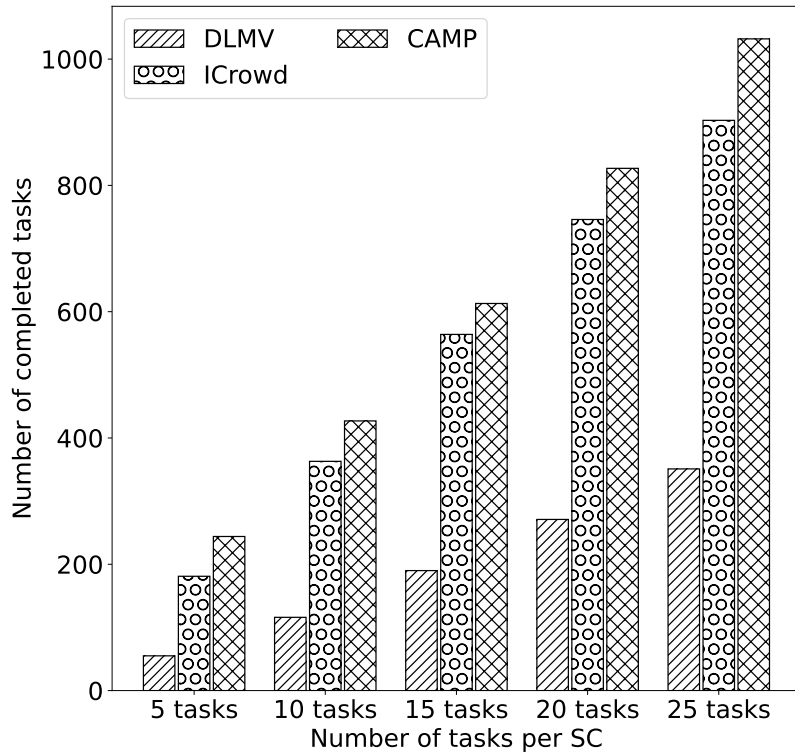
- total budget = budget per SC $|B|$ \times number of SC per day $|T|$ \times number of test days. Thus in this experiment, the total budget is 7500 (e.g., $25 \times 60 \times 5$).
- total number of tasks = number of tasks per SC \times number of SC per day $|T|$ \times number of test days. Thus in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

Figure 4.7 illustrates that the CAMP framework consistently maintains a remarkable performance advantage over both ICrowd and DLMV across all task settings. Specifically, in Scenario 1, CAMP achieves an average of 33.1 percent more completed tasks than ICrowd and an astounding 389.7 percent more than DLMV. Similarly, in Scenario 2, CAMP outperforms ICrowd by an average of 14.0 percent and DLMV by a substantial 219.7 percent in terms of completed tasks. These results underscore the remarkable effectiveness of the CAMP approach in optimizing task completion under varying task load scenarios in the context of MCS.

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



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



(b) Scenario 2: Different task map every SC

Figure 4.7: 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.

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.

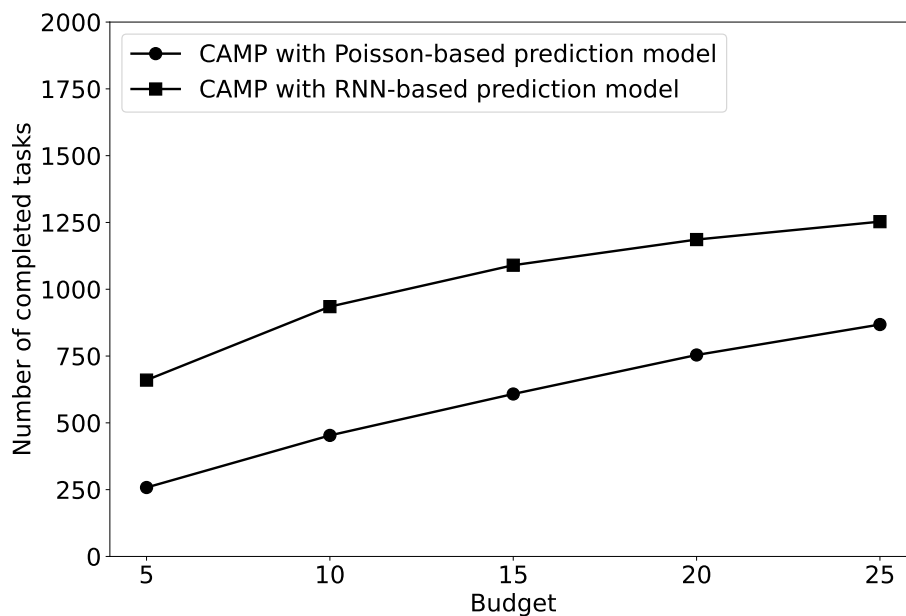
The presented results serve as compelling evidence of the CAMP framework's adaptability to varying task loads and reinforce its superiority over baseline methods. By effectively addressing the challenges posed by fluctuating workloads, the CAMP framework emerges as a robust solution that outperforms 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.

4.3.5 Effectiveness of the Prediction model

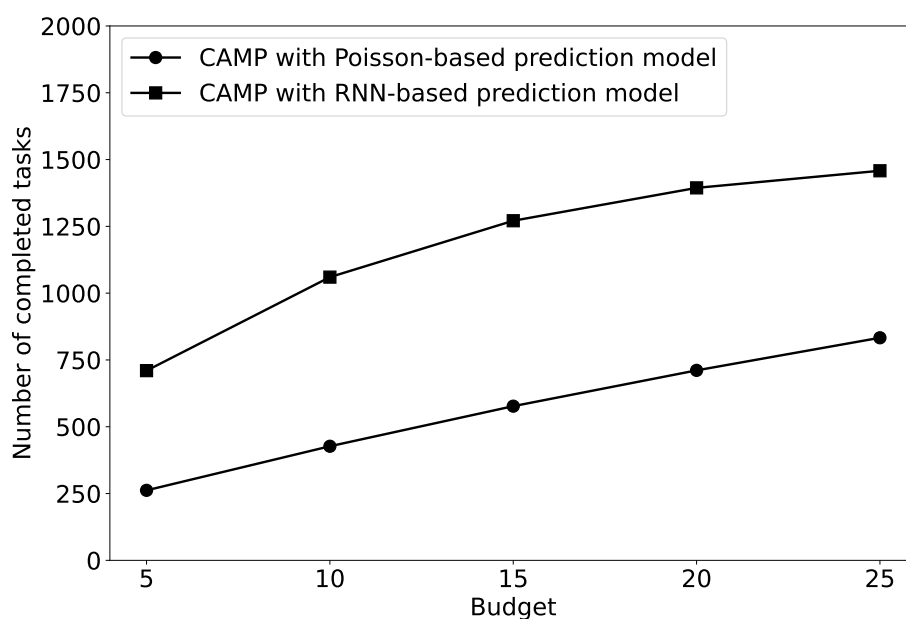
This subsection is dedicated to the evaluation of the prediction effectiveness in the first phase of the CAMP framework concerning OWs selection. The objective of this evaluation is to assess whether the RNN-based model contributes to an overall improvement in the performance and efficiency of the worker selection process.

Figure 4.8 provides a comprehensive comparison of how well the RNN-based model performed under various budget constraints and scenarios. The results consistently reveal that the RNN-based model surpasses the Poisson-based model across all budget scenarios. In Scenario 1, the RNN-based model achieves an average of 74.3 percent more completed tasks, underscoring its superior performance. Similarly, in Scenario 2, the RNN-based model outperforms the Poisson-based model by 109.7 percent, further substantiating its effectiveness.

This observed superiority of the RNN-based model can be attributed to several key factors. Firstly, the RNN-based model leverages the capabilities of recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) cells, which are well-suited for handling sequential data. In contrast, the Poisson-based model relies on a statistical process that may not accurately capture the complex and dynamic nature of human mobility patterns. The RNN-based model excels in predicting workers' future locations by analyzing their



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



(b) Scenario 2: Different task map every SC

Figure 4.8: Effectiveness of the RNN-based model for worker selection. 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.

historical mobility data. This accuracy plays a crucial role in the worker selection process, as more precise predictions lead to improved location-based decisions.

Secondly, the RNN-based model's effectiveness is reinforced by its ability to consider a worker's likelihood of visiting all locations when calculating utility, whereas the Poisson-based model does not account for location-specific task load. By prioritizing locations based on expected task load, the RNN-based model facilitates more efficient worker selection and optimized task allocation, resulting in higher task completion rates.

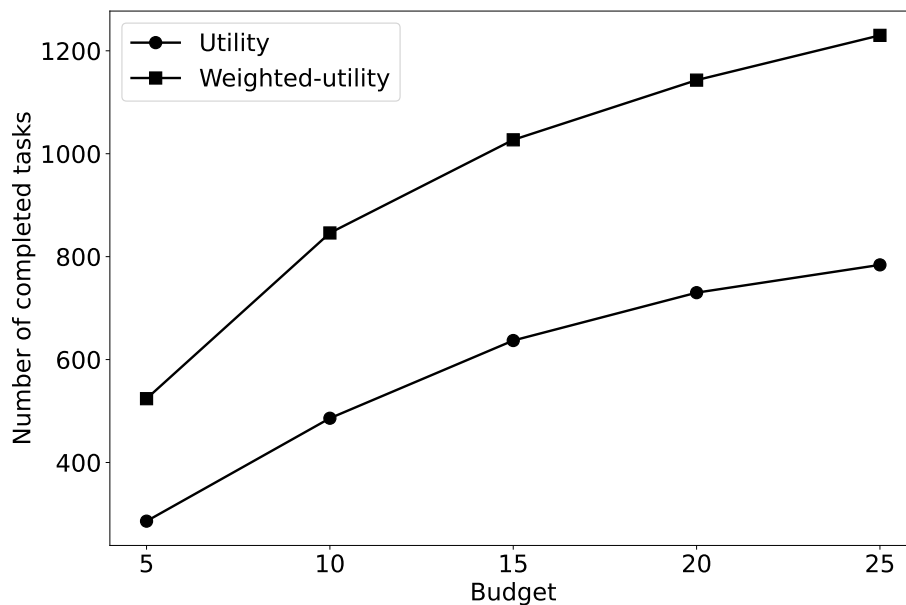
In conclusion, the comparative evaluation highlights the advantages of employing the RNN-based model in the CAMP framework. The RNN's capability to accurately predict worker mobility patterns, coupled with its strategic approach to worker selection, leads to enhanced task completion rates, making it a promising advancement in worker selection strategies for MCS applications. These findings underscore the practical value and potential impact of integrating the RNN-based model into real-world scenarios, where precise worker selection and improved task completion are essential considerations.

4.3.6 Effectiveness of the Weighted-utility Algorithm

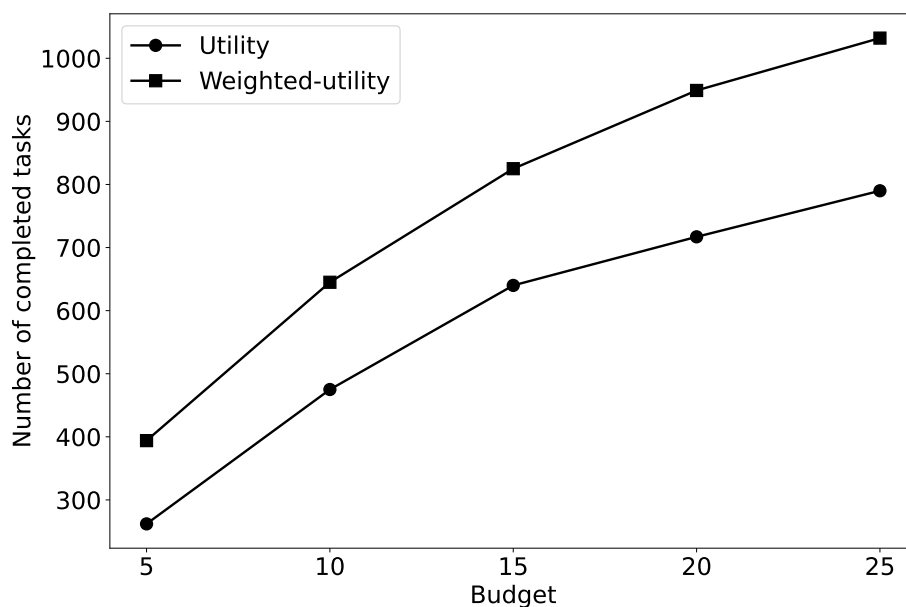
This subsection is dedicated to the evaluation of the weighted-utility algorithm's effectiveness in the second phase of the CAMP framework concerning OWs selection. The objective of this evaluation is to assess whether the weighted-utility approach contributes to an overall improvement in the performance and efficiency of the worker selection process.

Figure 4.9 provides a comprehensive comparison of how well the weighted-utility method performed under various budget constraints and scenarios. The results consistently reveal that the weighted-utility method surpasses the utility-based method across all budget scenarios. In Scenario 1, the weighted-utility approach achieves an average of 63.2 percent more completed tasks, underscoring its superior performance. Similarly, in Scenario 2, the weighted-utility algorithm outperforms the utility-based algorithm by 33.3 percent, further substantiating its effectiveness.

The primary advantage of the weighted-utility approach lies in its capacity to accurately prioritize locations based on the expected task load. This strategic feature enables the MCS system to select workers who are more likely to visit high-priority locations, resulting in an increased number of completed tasks within the given budget. By optimizing resource allocation, the system achieves higher task completion rates and overall enhanced performance. This evaluation unequivocally demonstrates the potential of the weighted-utility algorithm



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



(b) Scenario 2: Different task map every SC

Figure 4.9: 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.

to revolutionize the worker selection process within the CAMP framework. Embracing this approach in MCS systems holds the promise of elevating task completion rates, enhancing resource management, and augmenting system efficiency. It signifies a notable advancement in worker selection strategies for MCS applications with the potential to deliver substantial benefits.

4.4 Chapter Summary

This chapter introduces a two-phase framework called the CAMP, meticulously designed to tackle the intricate challenge of Opportunistic Worker (OW) selection within Mobile Crowd Sensing (MCS). CAMP's pivotal innovation is the seamless fusion of a highly precise multi-output Recurrent Neural Network (RNN) model for predicting worker mobility and an exclusive weighted-utility worker selection algorithm. This algorithm is acutely attuned to the dynamic task distribution across diverse locations and time slots, yielding a more refined and efficient worker selection process.

The rigorous evaluation, conducted with the real-world Crowdad Roma/Taxi dataset, provides compelling empirical support for the exceptional performance of the CAMP framework. Notably, CAMP surpasses baseline methods, namely iCrowd and DLMV, across both considered scenarios, whether employing the same task map or undergoing changes. This achievement is particularly significant, given the variable budget constraints and the fluctuating number of tasks generated in each sensing cycle (SC), underscoring CAMP's substantial potential to significantly augment worker selection efficiency across a spectrum of MCS applications.

In summary, the CAMP framework represents an innovative and highly efficient solution for optimizing the selection of OWs in the MCS domain. While acknowledging its current limitations and outlining pathways for refinement, CAMP's contributions lay a sturdy foundation for future advancements in worker selection strategies, promising heightened efficiency and effectiveness in diverse MCS applications, with the potential to reshape the landscape of MCS.

Chapter 5

Concluding Remarks

5.1 Summary of the Contribution

In summary, this dissertation addresses two core challenges within the domains of human mobility prediction and Mobile Crowd Sensing (MCS). These challenges are the development of accurate and efficient human mobility prediction models and the effective application of human mobility prediction to enhance MCS systems

The first framework introduced in this dissertation revolves around accurate human mobility prediction, a crucial element in enhancing various applications. It acknowledges that human mobility is a dynamic and non-linear process influenced by multiple factors. To bridge this gap, we present a two-phase framework that leverages spatio-temporal contexts from both the person of interest (POI) and their companions. This framework intelligently selects companions, overcomes the challenge of dimensionality through a stacked autoencoder, and employs a bidirectional recurrent neural network (BRNN)-based multi-output model for precise location predictions across multiple time slots. A weighted loss approach optimizes the prediction process by considering the importance of each future time slot.

The second framework, named 'Context-Aware Worker Recruitment based on a Mobility Prediction Model' (CAMP), shifts the focus to the selection of Opportunistic Workers (OWs) within the MCS context. CAMP, operating in two phases, efficiently selects the most suitable workers to maximize task completion within budget constraints. The first phase enhances worker mobility prediction through a recurrent neural network-based model, improving location prediction accuracy. The second phase introduces a weighted-utility worker selection algorithm, considering variations in task distribution. The integration of accurate mobility prediction and the weighted-utility worker selection algorithm positions CAMP as

a superior solution for optimizing task allocation within MCS.

Comprehensive experiments were conducted to evaluate the performance of these frameworks using diverse large datasets. The results consistently demonstrate the superior predictive accuracy of the location prediction frameworks and the ability of the worker selection framework to successfully complete more tasks within the same budget constraints compared to alternative approaches. These accomplishments underscore the potential of these frameworks to significantly enhance the accuracy and efficiency of human mobility prediction and worker selection, offering valuable contributions to the realms of predictive analytics and Mobile Crowd Sensing.

In summary, the main contributions of this dissertation are as follows:

- We begin by summarizing existing studies related to human mobility prediction models and worker selection in Mobile Crowd Sensing (MCS). Subsequently, we discuss the limitations of existing research. The central objective of this dissertation is twofold: to address the challenges of developing accurate and efficient human mobility prediction models and to effectively apply human mobility prediction to enhance MCS systems.
- Concerning the human mobility prediction model, we propose a multi-output Bidirectional Recurrent Neural Network (BRNN)-based model for predicting an individual's future locations based on recent mobility data from the Person of Interest (POI) and their companions. To combat the issue of high dimensionality, we introduce an autoencoder. Additionally, we utilize two companion selection methods to identify companions whose mobility data can contribute significantly to predicting the POI's future locations. Notably, we introduce a novel companion selection method based on a person ID embedding vector.
- In tackling the second challenge, we present the CAMP framework for the selection of Opportunistic Workers (OWs) in MCS projects. This framework incorporates an RNN-based multi-output human mobility prediction model that accurately forecasts worker locations in the next sensing cycle. The weighted-utility worker selection algorithm is employed to recruit a group of OWs capable of completing the maximum number of sensing tasks within budget constraints.
- We have conducted various experiments to evaluate the proposed human mobility prediction model and the worker selection model. Specifically, the location prediction

model is assessed using two large-scale Wi-Fi trace datasets, namely the UB and Dartmouth datasets. The results demonstrate the model's effectiveness in predicting an individual's future locations. The CAMP framework's performance is evaluated with a GPS-based taxi dataset, the Rome-taxi dataset. The results underscore CAMP's superiority over existing methods for OW selection in MCS, offering an effective solution for this important task.

5.2 Future Works

5.2.1 Enhancing Human Mobility Prediction

In the realm of human mobility prediction, the current research grapples with data limitations, hindering a comprehensive understanding of the intricate motivations behind individuals' movements. However, a commitment to advancement steers the direction of future work. The primary goal is to enhance the predictive models by delving deeper into the underlying reasons behind mobility patterns, transcending the mere prediction of destinations to gain insights into the driving forces behind these movements.

The forthcoming research initiatives involve a consideration of various pivotal factors aimed at refining predictive models. This includes exploring temporal dynamics to understand how mobility patterns fluctuate between workdays and weekends, influenced by daily routines that shape people's choices regarding their destinations. Additionally, the models will incorporate personal preferences, tailoring predictions to align with individual behaviors.

Recognizing the significant influence of environmental factors on mobility, we will also factor in variables such as weather conditions, traffic congestion, and the occurrence of special events. These elements play a substantial role in shaping individuals' mobility decisions, and their inclusion in the models will enhance prediction accuracy, providing a comprehensive understanding of how the environment intertwines with human mobility.

Looking forward, the integration of cutting-edge model architectures, such as attention mechanisms, transformers, and graph neural networks, will be a focal point of future research. These advanced architectures have shown promise in capturing intricate dependencies within sequential data and complex relationships in graph structures, which are inherent in human mobility patterns. By applying these state-of-the-art models, we aim to further improve the accuracy and interpretability of our predictions.

In summary, future research endeavors embrace a holistic approach, encompassing the se-

semantic context, temporal dynamics, personal preferences, and environmental factors within predictive models. This approach signifies a substantial step toward not only predicting individuals' future locations but also comprehending the motivations driving their movements, ultimately leading to more refined and accurate mobility predictions.

5.2.2 Expanding Worker Selection Framework

In the context of the worker selection framework, an existing limitation is the absence of consideration for various influential factors that can significantly affect worker selection. These factors include worker availability, skill levels, and past performance, among others. Recognizing this shortfall, future research endeavors hold promise in addressing this exciting avenue for exploration. A potential enhancement involves extending the weighted-utility worker selection algorithm to prioritize workers based on their availability, potentially leading to more efficient task completion, particularly when constrained by specific time limits.

Moreover, the broader applicability of the CAMP framework presents an intriguing opportunity for further validation. This can be achieved by subjecting the framework to rigorous evaluation using diverse real-world datasets and across a wide spectrum of Mobile Crowd Sensing (MCS) scenarios. This comprehensive approach promises to yield valuable insights into its performance and adaptability in various operational settings. By meticulously testing the proposed method in diverse contexts, we aim to acquire a holistic understanding of its strengths and identify potential areas for refinement. Ultimately, this pursuit charts the course towards establishing a more resilient and effective worker selection strategy for practical MCS implementations, serving a variety of applications and domains.

Bibliography

- [1] F. Asgari, V. Gauthier, and M. Becker, “A survey on human mobility and its applications,” *arXiv preprint arXiv:1307.0814*, 2013.
- [2] D. K. Rossmo, *Geographic profiling*. CRC press, 1999.
- [3] A. Wesolowski, N. Eagle, A. J. Tatem, D. L. Smith, A. M. Noor, R. W. Snow, and C. O. Buckee, “Quantifying the impact of human mobility on malaria,” *Science*, vol. 338, no. 6104, pp. 267–270, 2012.
- [4] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, “Location prediction on trajectory data: A review,” *Big data mining and analytics*, vol. 1, no. 2, pp. 108–127, 2018.
- [5] M. Chen, Q. Liu, W. Huang, T. Zhang, Y. Zuo, and X. Yu, “Origin-aware location prediction based on historical vehicle trajectories,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 1, pp. 1–18, 2021.
- [6] A. K. Laha and S. Putatunda, “Real time location prediction with taxi-gps data streams,” *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 298–322, 2018.
- [7] N. Meghanathan, “A location prediction based routing protocol and its extensions for multicast and multi-path routing in mobile ad hoc networks,” *Ad Hoc Networks*, vol. 9, no. 7, pp. 1104–1126, 2011.
- [8] L. N. Balico, A. A. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. Oliveira, “Localization prediction in vehicular ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2784–2803, 2018.
- [9] S. Siddharth, G. Tamilselvan, and C. Venkatesh, “Location prediction for improved human safety at complex environments,” *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 71, no. 3, pp. 5219–5234, 2022.

- [10] H. Bi, W.-L. Shang, and Y. Chen, “Cooperative and energy-efficient strategies in emergency navigation using edge computing,” *IEEE Access*, vol. 8, pp. 54 441–54 455, 2020.
- [11] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, “Next place prediction using mobility markov chains,” in *Proceedings of the first workshop on measurement, privacy, and mobility*, 2012, pp. 1–6.
- [12] D. Yao, C. Zhang, J. Huang, and J. Bi, “Serm: A recurrent model for next location prediction in semantic trajectories,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, Singapore, 2017, pp. 2411–2414.
- [13] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, “Predicting human mobility via variational attention,” in *Proceedings of the World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 2750–2756.
- [14] P. Ebel, I. E. Göl, C. Lingenfelder, and A. Vogelsang, “Destination prediction based on partial trajectory data,” in *Proceedings of 2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA: IEEE, 2020, pp. 1149–1155.
- [15] F. Alhasoun, M. Alhazzani, F. Aleissa, R. Alnasser, and M. González, “City scale next place prediction from sparse data through similar strangers,” in *Proceedings of ACM KDD Workshop*, 2017, pp. 191–196.
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [17] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, “CRAWDAD dataset dartmouth/campus (v. 2009-09-09),” Downloaded from <https://crawdad.org/dartmouth/campus/20090909>, Sep. 2009.
- [18] J. Shi, C. Qiao, D. Koutsonikolas, and G. Challen, “CRAWDAD dataset buffalo/phonelab-wifi (v. 2016-03-09),” Downloaded from <https://crawdad.org/buffalo/phonelab-wifi/20160309>, Mar. 2016.
- [19] H. Wang *et al.*, “A survey of application and key techniques for mobile crowdsensing,” *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–11, 2022.
- [20] S. Zhao, G. Qi, T. He, J. Chen, Z. Liu, and K. Wei, “A survey of sparse mobile crowdsensing: Developments and opportunities,” *IEEE Open Journal of the Computer Society*, vol. 3, pp. 73–85, 2022.

- [21] 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, 2023.
- [22] H. El Hafyani, M. Abboud, J. Zuo, K. Zeitouni, Y. Taher, B. Chaix, and L. Wang, "Learning the micro-environment from rich trajectories in the context of mobile crowd sensing: Application to air quality monitoring," *Geoinformatica*, pp. 1–44, 2022.
- [23] Y. Chen, W. Zhao, and C. Xu, "A survey on task allocation in mobile crowd sensing: Current state and challenges," in *Proceedings of the 2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Beijing, China, 2022, pp. 220–225.
- [24] T. Kandappu, A. Misra, and R. Tandriansyah, "Collaboration trumps homophily in urban mobile crowdsourcing," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, Portland, Oregon, USA, 2017, pp. 902–915.
- [25] 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 *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, San Francisco, California, USA, 2016, pp. 800–812.
- [26] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 1, pp. 1–33, 2014.
- [27] P. Cheng, X. Lian, L. Chen, and C. Shahabi, "Prediction-based task assignment in spatial crowdsourcing," in *Proceedings of 2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. San Diego, CA, USA: IEEE, 2017, pp. 997–1008.
- [28] 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. 68 111–68 125, 2022.
- [29] B. Altaf, L. Yu, and X. Zhang, "Spatio-temporal attention based recurrent neural network for next location prediction," in *Proceedings of 2018 IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA: IEEE, 2018, pp. 937–942.

- [30] S. Alemany, J. Beltran, A. Perez, and S. Ganzfried, “Predicting hurricane trajectories using a recurrent neural network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, Hawaii, USA, 2019, pp. 468–475.
- [31] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, “icrowd: Near-optimal task allocation for piggyback crowdsensing,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2015.
- [32] 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 Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4648–4659, 2020.
- [33] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, “CRAWDAD dataset roma/taxi (v. 2014-07-17),” Downloaded from <https://crawdad.org/roma/taxi/20140717>, jul 2014.
- [34] S. Jiang, G. A. Fiore, Y. Yang, J. Ferreira Jr, E. Frazzoli, and M. C. González, “A review of urban computing for mobile phone traces: current methods, challenges and opportunities,” in *Proceedings of the 2nd ACM SIGKDD international workshop on Urban Computing*, 2013, pp. 1–9.
- [35] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, “Understanding mobility based on gps data,” in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 312–321.
- [36] Y. Zhang, L. Wang, Y.-Q. Zhang, and X. Li, “Towards a temporal network analysis of interactive wifi users,” *EPL (Europhysics Letters)*, vol. 98, no. 6, p. 68002, 2012.
- [37] J. Fournet and A. Barrat, “Contact patterns among high school students,” *PloS one*, vol. 9, no. 9, p. e107878, 2014.
- [38] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, “Human mobility models for opportunistic networks,” *IEEE Communications Magazine*, vol. 49, no. 12, pp. 157–165, 2011.
- [39] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

- [40] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, “Wherenext: a location predictor on trajectory pattern mining,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France, 2009, pp. 637–646.
- [41] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, “Mining geographic-temporal-semantic patterns in trajectories for location prediction,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 1, pp. 1–33, 2014.
- [42] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, “Next place prediction using mobility markov chains,” in *Proceedings of the First workshop on measurement, privacy, and mobility*, Bern, Switzerland, 2012, pp. 1–6.
- [43] 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, 2019.
- [44] M. Chen, Y. Liu, and X. Yu, “Nlpm: A next location predictor with markov modeling,” in *Proceedings of Pacific-Asia conference on knowledge discovery and data mining*. Tainan, Taiwan: Springer, 2014, pp. 186–197.
- [45] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, “Limits of predictability in human mobility,” *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.
- [46] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, “Understanding individual human mobility patterns,” *nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [47] Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the next location: A recurrent model with spatial and temporal contexts,” in *Proceedings of the Thirtieth AAAI conference on artificial intelligence*, Phoenix, Arizona, USA, 2016.
- [48] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, “Gmove: Group-level mobility modeling using geo-tagged social media,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, CA, USA, 2016, pp. 1305–1314.
- [49] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [50] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

- [51] M. Granovetter, “Economic action and social structure: The problem of embeddedness,” *American journal of sociology*, vol. 91, no. 3, pp. 481–510, 1985.
- [52] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: user movement in location-based social networks,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1082–1090.
- [53] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg, “Inferring social ties from geographic coincidences,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 52, pp. 22 436–22 441, 2010.
- [54] J. L. Toole, C. Herrera-Yañe, C. M. Schneider, and M. C. González, “Coupling human mobility and social ties,” *Journal of The Royal Society Interface*, vol. 12, no. 105, p. 20141128, 2015.
- [55] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [56] D. Boston, S. Mardenfeld, J. S. Pan, Q. Jones, A. Iamnitchi, and C. Borcea, “Leveraging bluetooth co-location traces in group discovery algorithms,” *Pervasive and Mobile Computing*, vol. 11, pp. 88–105, 2014.
- [57] N. Eagle and A. S. Pentland, “Eigenbehaviors: Identifying structure in routine,” *Behavioral ecology and sociobiology*, vol. 63, no. 7, pp. 1057–1066, 2009.
- [58] C.-B. Nguyen, S. Yoon, and J. Kim, “Discovering social community structures based on human mobility traces,” *Mobile Information Systems*, vol. 2017, 2017.
- [59] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, D. Zhang, B. Guo, and Q. Lv, “Hy-tasker: Hybrid task allocation in mobile crowd sensing,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 598–611, 2019.
- [60] D. Deng, C. Shahabi, and U. Demiryurek, “Maximizing the number of worker’s self-selected tasks in spatial crowdsourcing,” in *Proceedings of the 21st acm sigspatial international conference on advances in geographic information systems*, Orlando, Florida, USA, 2013, pp. 324–333.
- [61] D. Deng, C. Shahabi, U. Demiryurek, and L. Zhu, “Task selection in spatial crowdsourcing from worker’s perspective,” *GeoInformatica*, vol. 20, pp. 529–568, 2016.

- [62] 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 Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2080–2093, 2020.
- [63] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 1, no. 1, pp. 1–28, 2015.
- [64] X. Chen, "A stable task assignment scheme in crowdsourcing," in *Proceedings of the 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. New York, USA: IEEE, 2019, pp. 489–494.
- [65] H. To, "Task assignment in spatial crowdsourcing: challenges and approaches," in *Proceedings of the 3rd ACM SIGSPATIAL PhD Symposium*, Burlingame, California, USA, 2016, pp. 1–4.
- [66] 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 of Things journal*, vol. 5, no. 5, pp. 3747–3757, 2018.
- [67] F. Yucel, M. Yuksel, and E. Bulut, "Coverage-aware stable task assignment in opportunistic mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3831–3845, 2021.
- [68] F. Yucel and E. Bulut, "Online stable task assignment in opportunistic mobile crowdsensing with uncertain trajectories," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 9086–9101, 2021.
- [69] J. Wang, J. Liu, and G. Zhao, "Two-phased participant selection method based on partial transfer learning in mobile crowdsensing," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–17, 2022.
- [70] J. Zhang and X. Zhang, "Multi-task allocation in mobile crowd sensing with mobility prediction," *IEEE Transactions on Mobile Computing*, vol. 22, no. 2, pp. 1081–1094, 2023.

- [71] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, Heidelberg, Germany, 2016, pp. 403–414.
- [72] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Seattle, Washington, USA, 2014, pp. 703–714.
- [73] J. Weinberg, L. D. Brown, and J. R. Stroud, "Bayesian forecasting of an inhomogeneous poisson process with applications to call center data," *Journal of the American Statistical Association*, vol. 102, no. 480, pp. 1185–1198, 2007.
- [74] I. O. Nunes, C. Celes, P. O. V. de Melo, and A. A. Loureiro, "Groups-net: Group meetings aware routing in multi-hop d2d networks," *Computer Networks*, vol. 127, pp. 94–108, 2017.
- [75] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2254–2265, 2012.
- [76] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Author's Publications

Patent

1. Seokhoon Yoon, **Quan T. Ngo**, "Method and Device for Transfer Learning-based Facial Expression Recognition Using Weighted Cluster Loss", Registered, Sep. 2022, Korean Patent No. 10-2441185.

International Journals

1. **Quan T. Ngo**, Seokhoon Yoon, "Facial Expression Recognition Based on Weighted-Cluster Loss and Deep Transfer Learning Using a Highly Imbalanced Dataset", *Sensors*, vol. 20, no. 9, p.2639, May. 2020.
2. **Quan T. Ngo**, Doi Thi Lan, Seokhoon Yoon, Woo-sung Jung, Taehyun Yoon, and Daeseung Yoo, "Companion Mobility to Assist in Future Human Location Prediction", *IEEE Access*, vol. 10, pp. 68111-68125, Jun. 2022.
3. Thi-Nga Dao, **Quan T. Ngo**, Cong-Binh Nguyen, Seokhoon Yoon, Jangyoung Kim, and Chunming Qiao, "A Low Cost Decentralized Future Contacts Prediction Model Using Wi-Fi Traces", *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 3807-3821, Nov. 2022.
4. **Quan T. Ngo**, Seokhoon Yoon, "Context-Aware Worker Recruitment for Mobile Crowd Sensing Based on Mobility Prediction", *IEEE Access*, vol. 11, pp. 92353-92364, Aug. 2023.

International Conferences

1. **Quan T. Ngo**, Seokhoon Yoon, "Facial Expression Recognition on Static Images", in *Proc. 2019 International Conference on Future Data and Security Engineering (FDSE)*, pp.640-647, Nov. 2019, Nha Trang, Vietnam.

2. **Quan T. Ngo**, Seokhoon Yoon, "Weighted—center Loss for Facial Expressions Recognition", in *Proc. 2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp.54-56, Oct. 2020, Jeju, Korea.

Domestic Conferences

1. **Quan T. Ngo**, Seokhoon Yoon, "Analysis of Encounters Between People using Wi—Fi Traces", in *Proc. 2018 The Korean Institute of Communications and Information Sciences Summer Conference*, pp.499-500, Jun. 2018, Jeju, Korea.
2. **Quan T. Ngo**, Dat Van Anh Duong, Doi Thi Lan, and Seokhoon Yoon, "Estimating Workers' Locations in Industrial Sites", in *Proc. 2021 The Korean Institute of Communications and Information Sciences Fall Conference*, pp.239-240, Nov. 2021, Yeosu, Korea.