A Human-Centered Cyber-Physical System Framework and its Applications in Gig Delivery

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yi Ding

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Tian He

October, 2022

Acknowledgements

Over the last six years, I have been fortunate to work with many people who guided me and supported me in this journey. First I want to thank my advisor Prof. Tian He, who gave me great guidance on how to do research and his passion for research has always been motivating me. I would also like to thank Prof. Desheng Zhang, who helped and supported me in my study and research, especially during difficult times. I have also been fortunate to work with Prof. Yunhuai Liu and Prof. Jie Gao, who share valuable suggestions on my work and career. I am also grateful to Prof. David Du, Prof. Feng Qian, and Prof. Yingling Fan for serving as the committee members of my preliminary exam, thesis proposal, and final defense. They provided valuable feedback that significantly improved the quality of the dissertation.

The best memories of my Ph.D. study are definitely the experience with researchers and students from different groups, including Dr. Shuai Wang, Dr. Zhimeng Yin, Dr. Wenchao Jiang, Dr. Song Liu, and Dr. Ruofeng Liu from MiNDs Group at the University of Minnesota; Dr. Yu Yang, Dr. Guang Wang, Guang Yang, Zhiqing Hong, and Wenjun Lyu from ARSENAL Group at the Rutgers University; Dongzhe Jiang, Yan Zhang, and Zhigang Dai from Peking University; Baoshen Guo and Wei Liu from Southeast University. I will not forget the excellent works we have done together as well as the happy time we have spent.

During my work at the Alibaba Group in Shanghai, I also received support from many people including Hao Zhang, Ling Liu, Yun Ji, Qiao Chen, Yibin Sui, Xingxiang Zhuo, Dr. Guobin (Jacky) Shen, Dr. Pengfei Zhou, Dr. Yang Li, Dr. Yan Zhang, Dr. Hao Xia, Dr. Jing Zhao, and Fan Ding. Their support and help mean a lot to me in my work.

I also want to thank Prof. Mats Heimdahl and the staff members from the Department of Computer Science and Engineering at the University of Minnesota for their support in my study.

Lastly, I want to express my sincere gratitude and love to my family and friends. My parents have always been supporting me on my back. I also received encouragement from many friends, especially Yicun Wang, Dongxu Zheng, Dongdong Wang, Xiaoxiao Sun, and Yuyuan Liu. My two cats, Pipi and Datou also play key roles in completing my research works. At last, I want to thank Miao Yu, my wife, who share all the sorrow and happiness along the journey with me.

Dedication

To Miao, My Lifelong Lover, Partner, and Friend.

Abstract

With wider and deeper interaction between humans and systems in modern society, the study of human-centered cyber-physical systems (human-centered CPS) has become increasingly important. Thanks to the massive data collected by ubiquitous devices (e.g., smartphones) and advanced machine learning and data mining techniques, numerous human-centered CPS applications and studies are emerging. However, two essential problems still exist: (1) unlike purely internet-based systems, in human-centered CPS, different people engage the system at different places using different devices, which brings technical challenges like scalability and heterogeneity; (2) unlike CPS without wide and deep human participation, in human-centered CPS, human behavior (e.g., locations, mobility, activity) plays a key role, but human behavior is difficult to predict given its inherent uncertainty. To address the challenges, we have done a variety of works that can be organized under the three-layer framework of sensing, prediction, and decision-making. In the sensing layer, we design and build wireless sensing systems to capture human behavior like the arrival and departure at certain locations. We address the scalability challenge by studying human mobility and adopting their smartphones as virtual sensors, and we address the heterogeneity challenge by studying the impacts of environment and hardware on sensing and modeling the similarity with graph learning. In the prediction layer, we study the indoor localization problem by transforming it into a travel time prediction problem and solving it with graph learning based on the human behavior data collected from wireless sensing. In the decision-making layer, we utilize the data from the sensing and the knowledge from prediction to make decisions that lead to higher efficiency compared to the state-of-the-art. We also show how to utilize the feedback from humans to benefit the system design and achieve human-system synergy. In addition to the in-lab design and experiments, we implement our works in one of the latest and largest human-centered CPS applications, gig delivery. By studying couriers' and merchants' behavior and building corresponding sensing, prediction, and decision-making systems, we not only improve the system performance but also achieve the synergy between the couriers and systems, saving millions of dollars for the platform and benefiting millions and merchants, couriers, and customers.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cyber-Physical System (CPS) is the integration of computation with the physical process. The cyber part (e.g., computers) first monitors the physical process by sensing the signals, then do some predictions based on the data collected, and finally produces some actions based on the knowledge in the models, and gives the actions to the physical process. Many real-world systems can be modeled and studied in the CPS scheme, such as industry 4.0 and smart grid. Meanwhile, there are some systems that interact directly and deeply with human, where the system acts on the human and vice versa, such as location-based social networks, gig delivery, and intelligent transportation. These systems can be defined as the *human-centered CPS*.

A key problem in human-centered CPS is "How to achieve synergy between humans and systems in the future with advanced AI and computing infrastructure enriching the interaction between humans and systems exponentially". This raises research problems in two directions: (1) how to build better systems with human input, that is, human help systems, and (2) how we can better serve human with the system, that is, system help human In recent years, there are extensive investments from both the public sector and the private sector for Human-Centered CPS.

Compared to other systems without explicit and wide human participation, there are some new opportunities for building human-centered CPS. A key opportunity is the human behavior, such as human mobility and activity. Human behavior works as a special signal and can help us understand the human and the physical world. More importantly, with ubiquitous devices (e.g., smartphones, smartwatches, IoT devices) around, there are cost-effective ways to collect information about human behavior. Based on the behavior data collected, we can also build behavior models to predict their future behavior. Based on future behavior, some actions such as notification and recommendation, impact human behavior as proactive intervention. That is, we have a close loop for human behavior, and has the potential to achieve human-system synergy, where systems help humans, and humans also help systems.

Although the study of human behavior has the potential to build better systems, there are many challenges we have to address, which can be categorized as human-related and system-related. For human-related challenges, one important challenge is the uncertainty in human behavior. For example, human behavior is diverse, i.e. different people may have different behavior even in the same situation. Moreover, human mobility and activity pattern is not fixed and changes with time. Even we can model human behavior considering the diversity and dynamic factors, there is still some randomness in human behavior, especially at the individual level in short term, making it difficult to study. For system-related challenges, cost and heterogeneity are two major challenges when we are talking about large-scale systems in the wild. For example, to build a sensing system for a large-scale (e.g., nationwide) commercial delivery platform, it's almost impossible to deploy high-cost sensors at each site and assume the environments in each site are the same. We need to study and address the challenges before we can build human-centered CPS in the real world.

In the dissertation, I will introduce how we build human-centered CPS through innovative works in sensing, prediction, and decision-making. Each work is an individual topic, and together they organize as different layers for a complete human-centered CPS, specifically, a real-world nationwide gig delivery platform. In the **sensing** layer, I will introduce the Bluetooth sensing system we designed and deployed to detect couriers' arrival at the merchant. We utilize the Bluetooth function on couriers' and merchants' smartphones to make the system low-cost, energy-efficient, and compatible [1, 2]. We also use the advanced graph learning techniques to solve the environment and smartphone heterogeneity in the wild [3]. In the **prediction** layer, I will introduce how we infer couriers' indoor locations by investigating spatiotemporal information from their encounter events [4]. We use Bluetooth signals to capture their encounter events and use graph learning to predict the travel time from their encounter locations to the nearby merchants. In the **decision-making** layer, I will use two examples to illustrate how we achieve human-system synergy. For the problem of building a better system with human input, I will introduce how we notify couriers of their early arrival and use their feedback to improve the arrival detection system [5]. For the problem of better serving humans with systems, I will introduce how we conduct better order scheduling to save couriers' delivery time. In particular, the contributions in this dissertation are as follows:

- We conducted a detailed study on how to integrate human behavior – especially their locations and mobility – in the CPS with proper sensing, prediction, and decision-making techniques. In collaboration with a gig delivery company in China, we implement and evaluate our work by building a nationwide Bluetooth sensing system, serving more than 1 million couriers, 3 million merchants, and 186 million customers.

- In the sensing layer, we address the environment and device heterogeneity problem in the wild by utilizing the similarity in the influence patterns of different environments and smartphones. We designed `Para-Pred` [3], a sensing framework based on graph neural networks, which includes: (1) a similarity extraction module to capture the influence patterns of different environments and smartphone models and explore the potential similarity information. (2) a parameter prediction module that effectively combines the similarity information to predict accurate status estimation models for unseen scenarios. The framework is evaluated on the data collected from aBeacon [1], a citywide Bluetooth sensing system we built in Shanghai, China, including 109,378 couriers with 672 smartphone models in 12,109 merchants. Experiments show that we outperform the state-of-the-art methods by 11.87%, 14.54%, 10.33%, and 12.27% in accuracy, recall, precision, and F1-score, respectively.

- In the prediction layer, we designed $\texttt{P}^2\texttt{-Loc}$ [4] to infer couriers' locations without extra infrastructure costs based on the ubiquitous encounter events among couriers and their indoor mobility preferences. Specifically, we build a graph to implicitly learn the topology of indoor merchants from couriers' historical indoor travel data, then we use a graph neural network (GNN) to integrate node information and topological structure of the graph and use link prediction to predict couriers' travel time to all the merchants. We implement and evaluate $\mathtt{P}^2$-`Loc` in a mall with 4,075 couriers and 79 merchants for a month. The results show that we outperform baselines with Wi-Fi, GPS, and manual reporting by 9%, 19%, and 51%, respectively, and outperform other encounter-based methods (i.e., MDS-based and statistical) by 8% and 31%.

- In the decision-making layer, for humans to help systems, we designed an early arrival notification module based on the sensing and prediction system we have built to get couriers' feedback regarding their arrival reports and use their feedback to improve the sensing and prediction performance. Data show that the incorrect notification is reduced by 25% based on feedback from couriers. For systems to help humans, we conduct a simulation based on real-world data to study the benefit of considering fine-grained couriers' behavior (e.g., mobility, locations) in order scheduling. The results show that using fine-grained localization data can reduce the platform's overdue rate to save $40,000 every day.

The remaining part of the dissertation is organized as follows. The state-of-the-art related to human-centered CPS is introduced in Chapter 2, followed by the framework of this dissertation in Chapter 3. Chapter 4 introduces some background of gig delivery and how we build a real-world large-scale sensing system for human behavior and address the underlying challenges. Chapter 5 introduces how we conduct human behavior prediction. Chapter 6 use two examples to show how the decisions from sensing and prediction can help humans and systems achieve synergy. We discuss our future work in Chapter 7 and conclude the dissertation in Chapter 8.

Chapter 2

State-of-the-Art on Human-Centered CPS

2.1 Overview

The study of human-centered CPS is at the intersection of system, data, and social computing. Given the strong connection of CPS and IoT [6], the study of CPS is rooted in wireless sensing and mobile computing where we can collect data from pervasive devices like smartphones and develop interesting applications around human behavior like indoor localization [7], transportation model detection [8], and breathing monitoring [9]. With more and more data collected from ubiquitous devices and the advances in data mining and machine learning techniques, data-centric solutions (i.e., data mining and machine learning) start to play key roles in sensing, prediction and decision-making related to human behavior, such as mobility prediction [10], delivery time prediction [11], vehicle routing [12], and order dispatching [13]. Unlike other CPS applications driven by the studies of robots and computers, human-centered CPS also calls for social computing studies to understand the impact of humans on the systems and vice versa. Recent works focus on equity and well-being of gig workers [14, 15], and human-AI collaboration [16].

In the following sections, I will summarize the works related to human-centered CPS from the perspective of wireless sensing and mobile computing, data mining and machine learning, and social computing. All of these works are indispensable to understanding, studying, and building human-centered CPS to be effective, efficient, and socially aware.

2.2 Wireless Sensing and Mobile Computing Perspective

The related works in wireless and mobile computing can be categorized in two orthogonal dimensions: the underlying sensing or communication technologies (i.e., Wi-Fi, Bluetooth) and the sensing target or applications (i.e., locations, activities).

For sensing and communication technologies, novel sensing solutions are designed based on a variety of technologies that can be categorized as (1) communication-related technologies like Wi-Fi [17], Bluetooth [18], LoRa [19], and cellular [20]; (2) physical-signal-related technologies like acoustic [21], LED light [22], barometers [23], inertial measurement unit (IMU) [24], and magnetic [25]; (3) dedicated sensing technologies like GPS [26], RFID [27], ultra-wideband (UWB) [28], radar [29], pyroelectric infrared (PIR) sensors [30], and smartphone camera [31]. Many of the technologies have been integrated into ubiquitous devices like smartphones and smartwatches, while some have the potential to be integrated into other devices such as drones [28] and vehicles [32]. In many cases, different technologies can be utilized for the same sensing task (e.g., GPS, Wi-Fi, Bluetooth for localization), so customized solutions are needed to take the advantages and disadvantages of different technologies into consideration for specific applications. For example, GPS, Wi-Fi, and Bluetooth provide different granularities of indoor location information at different costs, as we will discuss in detail in Chapter 4. With more and more sensing technologies developed and integrated into the smartphones like Lidar and 3D cameras, there will be more and more sensing solutions in the future for human behavior.

For sensing targets and applications, novel sensing solutions are designed for a variety of targets and scenarios that can be categorized as (1) mobility-related applications like locations [7], presence [33], transportation mode [8], driving [34]; (2) activity-related applications like general activity [35], gesture [36], handwriting [37], drinking [38], eye-blink [39]; (3) health-related applications like emotion [40], heart rate variability (HRV) [31], breathing [9], Parkinson's [41], and Alzheimer [42]; (4) interaction-related applications like proximity [43], human-object interaction [44], and social interaction [45];

(5) context-related applications like ambient sound [46], nightlife patterns [47], and local business ambience [48]. Applications are usually driven by the newly integrated sensors on smartphones or the advanced data mining and machine learning techniques to build models for sensing tasks. In the future, new sensing targets and applications will continue to emerge based on not only the sensing hardware and machine learning techniques but also the achievements from other fields such as health, psychology, and social science.

2.3 Data Mining and Machine Learning Perspective

As in wireless sensing and mobile computing, the related works in data mining and machine learning can also be categorized in two dimensions: the data mining and machine learning techniques utilized (i.e., graph learning, reinforcement learning), and the learning target or applications (i.e., travel time, order dispatching).

For data mining and machine learning techniques, the research around human-centered CPS keeps absorbing the advanced learning techniques such as binary classification [42], clustering [49], long short-term memory (LSTM) [50], recurrent neural network (RNN) [51], convolutional neural network (CNN) [52], graph convolutional network (GCN) [53], meta-learning [54], generative adversarial network (GAN) [55], reinforcement learning (RL) [56], and federated learning [57]. Some techniques were first proposed for problems in other fields (e.g., CNN for image processing), but with some modifications and customization, the core idea and the model structure can usually be applied to solve problems in human-centered CPS.

For learning targets or applications, in additional to the direct sensing targets and applications discussed in Section 2.2, there are more applications that can be explored based on the data related to human behavior and mobility, which can be categorized as prediction-oriented and decision-making-oriented. For prediction-oriented works, there are works in different applications like food preparation time [58] and delivery time [59] in food delivery; travel time [60], and demand [61] in ride-hailing; delivery time [11] and service time [62] in express; and passengers' mobility [10], parking availability [50], map generation [63], and urban function discovery [64] in smart cities. For decision-making-oriented works, there are also works like courier dispatching in taxi [65], express [56], and on-demand ride-hailing [66]; bike lane planning [67] and re-balancing [49] in bike sharing; and vehicle routing [12], demand prediction [52], transportation recommendation [68], and electric-car charging scheduling in smart cities. The explosive growth of applications is driven by two factors: the data can be collected (e.g., GPS) and the problems not well solved (e.g., dispatching). These two factors will still exist in the future and drive more and more applications in this direction.

2.4 Social Computing Perspective

In the previous several years, I have been mainly working on specific human-centered CPS applications like gig delivery and express, which are related to several broad topics like the gig economy and crowdsourcing in the social computing community. The related works can be categorized based on the focuses of the research: (1) humans in the system such as bias [14], interaction [69], collaboration [70], social isolation [71], emotional support [15], skill provision [72], risk-sharing [73]; (2) relation between humans and the systems such as tensions and alliances [74], dissonance [16], algorithmic fairness [75]; (3) system with human participation such as participatory sensing [76], mobile crowdsourcing [77], rider sharing built by grassroots [78]. Although almost all computer systems are designed for humans, the problem becomes complicated when there are multiple groups of humans in the system (e.g., customers, merchants, couriers, and platform stuff in a food delivery platform) because sometimes different groups have contradictory goals. The study of social computing allows us to guarantee the welfare for different groups such as fairness and well-being for couriers, which may become increasingly important in future human-centered CPS applications where sensing techniques and AI are deeply involved in our everyday work and life.

Chapter 3

Framework

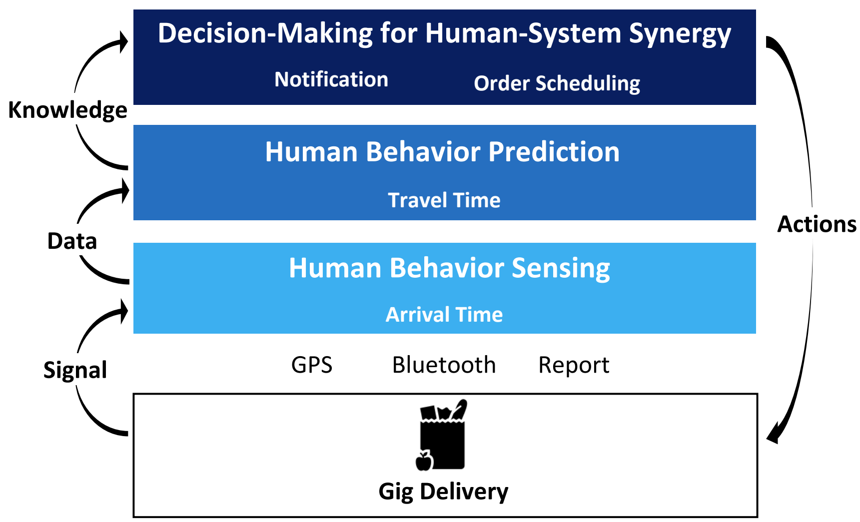


Figure 3.1: Framework

In this chapter, we introduce the framework of this dissertation. As shown in Figure 3.1, the framework consists of three components, including human behavior sensing, human behavior prediction, and decision-making for human-system synergy.

In the sensing part (Chapter 4), we designed and deployed a low-cost citywide physical beacon system (`aBeacon`), and based on that designed and deployed a nationwide infrastructure-free virtual beacon system (`VALID`) to detect couriers' arrival at the merchants in the wild. Serving more than 3 million merchants and 1 million couriers in more than 300 cities in the wild. Then we employ an indoor status estimation system (i.e., `Para-Pred`) based on the similarity information to address environment and smartphone model heterogeneities for large-scale scenarios. We evaluate `Para-Pred` on the data collected from real-world `aBeacon`system. Experiments show that we outperform the state-of-the-art methods by 11.87%, 14.54%, 10.33%, and 12.27%, in accuracy, recall, precision, and F1-score, respectively. In addition, we analyze the effectiveness of the similarity extraction module and the attention mechanisms. We also study the performance sensitivity of our system to different shop types and smartphone models.

In the prediction part (Chapter 5), based on the sensing techniques introduced in Chapter 4, we introduce $\texttt{P}^2\texttt{-Loc}$, an infrastructure-free indoor localization system in on-demand delivery to predict the travel time from couriers' current locations to nearby merchants' locations. We build a graph to implicitly learn the topology of indoor merchants from couriers' historical indoor travel data, then we use GNN to integrate node information and the topological structure of the graph and use link prediction to predict couriers' travel time to all the merchants. We prototype and implement $\texttt{P}^2\texttt{-Loc}$ on a commercial on-demand delivery platform, and evaluate $\texttt{P}^2\texttt{-Loc}$ in a mall with 4,075 couriers and 79 merchants for a month. The results show that $\texttt{P}^2\texttt{-Loc}$ outperforms methods based on Wi-Fi, GPS and reporting by 9%, 19%, and 51%, respectively, and outperforms other encounter-based methods (i.e., MDS-based and statistical) by 8% and 31%.

In the decision-making part (Chapter 6), we will use two examples to show (1) how we build an early arrival notification mechanism to help improve the system performance with the feedback from couriers based on the beacon system introduced in Chapter 4; (2) how we apply the results from travel time prediction to improve the order dispatching efficiency based on the $\texttt{P}^2\texttt{-Loc}$ introduced in Chapter 5.

Chapter 4

Arrival Detection with BLE Beacon in Heterogeneous Scenarios

4.1 Introduction

Nowadays, gig delivery is an emerging business for Gig Economy [79], where Gig workers deliver online orders (e.g., food) within a short time (e.g., 30 minutes) from merchants (e.g., restaurants) to customers. This business proliferates with the emergence of several delivery platforms worldwide, e.g., Prime Now [80], UberEats [81], Instacart [82], and DoorDash [83] in the U.S.; Deliveroo [84] in the U.K.; Meituan [85] and Eleme (i.e., Alibaba local service company, our collaborated platform) [86] in China.

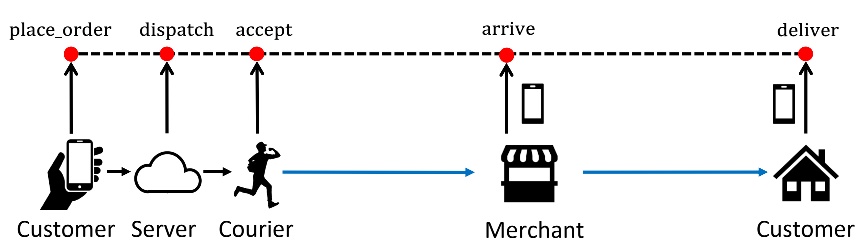


Figure 4.1: Gig Delivery Process

In general, the delivery service can be simplified as a process from placing the order to the final delivery (Figure 4.1). In this process, a customer places the order, and then the server dispatches the order to a courier. After accepting the order, the courier moves to the merchant to pick up the order. When arrives at the merchant, the courier will report arrival on the APP. Finally, the courier delivers the order to the customer and reports the delivery. For the platform, it is essential to know couriers' real-time arrival status at merchants, which is used to (1) update order status in customer's APPs for better customer experience, (2) assign new orders to the most suitable couriers, and (3) train learning models to estimate the order's preparation and delivery time for future orders [87].

Admittedly, indoor arrival and departure status detection is not technically challenging and has been widely investigated in controlled environments, e.g., labs, museums, and airports. However, it is still an open question for city-wide detection in the wild. In industry, solutions mainly rely on either the courier's smartphone GPS (which is inaccurate in indoor environments and cannot reveal floor information) [88] or manual reporting (which suffers from intentional or unintentional human errors). In academia, the solutions are based on Wi-Fi [89, 90, 91, 92, 93], LED fixtures [94, 95, 96], RFID [97, 98, 99], or natural landmarks [100, 22, 25, 101]. However, each of them has limitations for a city-wide or even nationwide deployment with millions of merchants and couriers with only commodity smartphones. Wi-Fi-based solutions have three limitations: (1) continuous scanning is required to keep the Wi-Fi list updated, which brings much extra power consumption for courier's smartphones; (2) for merchants without Wi-Fi Access Point devices, it is costly to deploy new ones [102, 103]; (3) ordinary APPs have no access to the Wi-Fi list on iOS devices. LED solutions do not scale up due to hardware modification costs [94]. RFID solutions require additional equipment on both receivers and transmitters. Some infrastructure-free methods utilize natural landmarks like sound [100], light [22], and magnetic [25]. Still, they all need a labor-intensive fingerprinting process and are prone to interference, which is uncontrollable in the wild.

Compared to the existing solutions, Bluetooth Low Energy (BLE) [18, 104, 105, 106] is a promising solution to achieve a better trade-off of accuracy and scalability for the arrival detection task. From 2017 to 2021, in collaboration with the Eleme company in Alibaba Group, one of the largest gig delivery platforms in China, we designed and deployed two BLE-based beacon systems, the physical beacon system (`aBeacon`) [1], and the virtual beacon system (`VALID`) [2]. `aBeacon` consists of customized hardware BLE devices at 12,109 merchants, interacting with 109,378 couriers to infer their arrival status to assist the scheduling of 64 million delivery orders for 7.3 million customers in its full operation stage (2018/01-2020/04). `VALID` consists of virtual beacons (i.e., Existing APP on merchants' smartphones) at 3 million shops and restaurants, to infer and influence 1 million couriers' behavior and assist the scheduling of 3.9 billion orders for 186 million customers in 364 Chinese cities in its full operation stage (2018/12-2021/01). With the BLE-based beacon systems, we can build an indoor status estimation model to infer the accurate arrival and departure time of couriers at the merchants.

However, it has been observed that many factors in the wild impact the performance of the BLE-based beacon systems. First, a small variation in the hardware and software of smartphones may cause significant heterogeneity in the collected sensor data. Different smartphones respond differently even to the same status [107]. Second, the propagation path of Bluetooth signals is impacted by locations where beacon devices are deployed and surrounding environments [108]. These factors affect Bluetooth signals in different ways simultaneously, which lead to varied collected data from smartphones, even for the same smartphone. Thus, it is very challenging to use an indoor status estimation model trained with data from one scenario directly in other scenarios due to the heterogeneity.

Some approaches have been proposed to address the environment and smartphone model heterogeneities in wireless sensing. For the smartphone model heterogeneity, conventional solutions utilize reliable but expensive sensors, which are unsuitable for large-scale deployment. Furthermore, researchers apply deep neural networks to learn the awareness of smartphone model heterogeneity [109, 110]. For the environment heterogeneity, recent solutions leverage transfer learning [111, 112] or adversarial learning [113, 114] to suppress the influence of environmental heterogeneity on recognition tasks. These methods are difficult to be applied in city-wide scenarios with thousands of merchants and hundreds of smartphone models, which require re-collect a large amount of labeled data or re-train models when dealing with new scenarios.

To address the environment and smartphone heterogeneity in large-scale real-world scenarios, we propose `Para-Pred`, an indoor status estimation framework based on graph neural networks. We first construct a shop-phone interaction graph considering shops and smartphone models as nodes, and historical Bluetooth signal data as edge information, to learn the influence patterns of the heterogeneities by the environment heterogeneity model and smartphone heterogeneity model. Then, based on the similarity between these influence patterns, we construct a shop-shop similarity graph and a phone-phone similarity graph. We design a parameters prediction module based on the similarity between the influence patterns of the heterogeneities to predict the indoor status estimation model parameters for unseen scenarios. Finally, we obtain effective status estimation models for unseen scenarios which can be used directly for accurate status estimation without additional data collection and model retraining. In summary, our key contributions are as follows:

- We designed and deployed a low-cost citywide physical beacon system (`aBeacon`), and based on that designed and deployed a nationwide infrastructure-free virtual beacon system (`VALID`) to detect couriers' arrival at the merchants in the wild. Serving more than 3 million merchants and 1 million couriers in more than 300 cities in the wild. Two data sets are released to the community for future research [115, 116].

- We employ an indoor status estimation system (i.e., `Para-Pred`) based on the similarity information to address environment and smartphone model heterogeneities for large-scale scenarios. Based on the similarity information between influence patterns of the heterogeneities, we directly infer the effective indoor status estimation model for unseen scenarios without additional data collection and model retraining.

- We evaluate `Para-Pred` on the data collected from real-world `aBeacon`system. Experiments show that we outperform the state-of-the-art methods by 11.87%, 14.54%, 10.33%, and 12.27%, in accuracy, recall, precision, and F1-score, respectively. In addition, we analyze the effectiveness of the similarity extraction module and the attention mechanisms. We also study the performance sensitivity of our system to different shop types and smartphone models.

4.2 Bluetooth Beacon System

Compared to other sensing and localization solutions (e.g., Wi-Fi, GPS, landmarks, RFID), Bluetooth has the potential to achieve a better trade-off between accuracy and scalability for arrival detection in the wild. In 2017, we designed a hardware-based physical beacon system and deployed the system to 12,109 merchants in Shanghai by collaborating with a gig delivery company in China. After analyzing the advantages and disadvantages of the physical beacon system, we designed a software-based virtual beacon system and deployed it to more than 3 million merchants in China by collaborating with the same company. In this section, we introduce the two beacon systems as the infrastructure for the following sensing, prediction, and decision-making tasks. We will also discuss the heterogeneity in the beacon system, one of the major challenges in the large-scale sensing systems in the wild.

4.2.1 Physical Beacon System

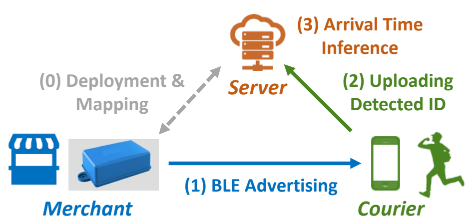


Figure 4.2: Physical Beacon Framework

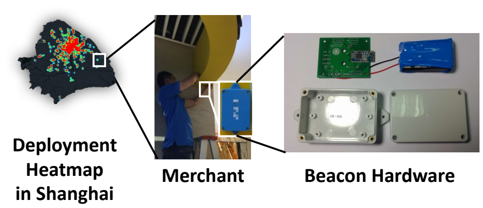


Figure 4.3: Physical Beacon Deployment

As shown in Figure 4.2, in the physical beacon system `aBeacon`, a generic workflow is as follows: (1) devices deployed in indoor merchants to continually broadcast their ID tuples; (2) an embedded BLE scanning module in the Alibaba couriers' smartphone APP (mandatory for all couriers) to receive these ID tuples from devices when in proximity and to upload them to a back-end server using the smartphone Internet connectivity; (3) The server updates couriers' arrival and uses them for various functions, e.g., new order scheduling.

As shown in Figure 4.3, we deployed 12,109 customized `aBeacon` devices to 12,109 merchants on Alibaba platform in Shanghai. An `aBeacon` device is a low-cost ($8) broadcast-only BLE device. It does not have GPS or cellular/ Wi-Fi connections, so it cannot receive any update, and it also cannot directly communicate with back-end server.

4.2.2 Virtual Beacon System

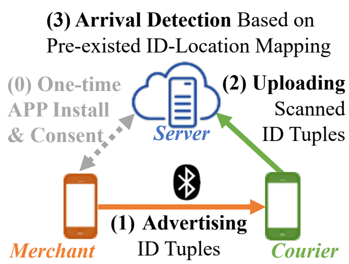


Figure 4.4: Virtual Beacon Framework

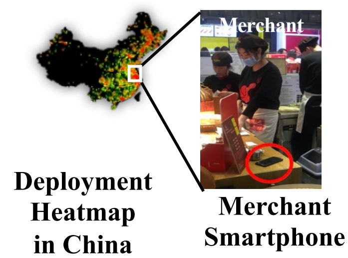


Figure 4.5: Virtual Beacon Deployment

As in Figure 4.4, in the virutal beacon system `VALID`, we (0) ask for merchants' consents that we can use their smartphones to detect couriers when they install our merchant APP; (1) let consenting merchants' phones work as virtual beacons to advertise `ID tuples` continuously by the BLE 5.0 protocol when they are in the order accepting status; (2) let couriers' phones passively scan for `ID tuples` and then upload received `ID tuples` to a back-end server in real-time by Internet connection (e.g., cellular); (3) let the server with the received `ID tuples` check a pre-stored mapping between `ID tuples` and merchant IDs to detect an arrival finally.

As shown in Figure 4.5, we deployed and operated `VALID` nationwide from 2018/12 to 2021/01. It is deployed to infer the indoor arrival status of 1 million professional couriers at 530,859 indoor merchants in 364 Chinese cities and few of them, if any, are under our control, i.e., in the "wild".

4.2.3 Follow-Up Works

Based on the physical beacon and virtual beacon systems, there are some follow-up works that either solve the challenges that emerged in the systems (e.g., heterogeneity) or build upper-layer applications (e.g., indoor localization, location correction). In the following, we will discuss the impact of device and environment heterogeneity in the beacon system and how we address it [3]. We have also built some applications. For example, in SmartLoc [117], we build an indoor localization system based on the virtual beacon system; in ALWAES [118], we build a location correction system based on both physical and virtual beacon systems. The physical and virtual beacon systems also work as ground truth for some related works such as IODetecotr [119], an indoor/outdoor detection system, and WePos [120], a Wi-Fi-based indoor localization system.

4.3 Heterogeneities in Beacon System

Although the physical and virtual beacon systems provide accurate arrival data for the upper layer applications and improve the overall all efficiency of the order scheduling system, there are some challenges to building accurate arrival departure detection models. One of the major challenges is the heterogeneity of the deployment environment and the couriers' smartphone hardware since the systems are deployed to millions of shops and interact with millions of couriers. In the following, we analyze the heterogeneities in the real beacon system to show the impact of heterogeneities on the system reliability and arrival detection model performance.

4.3.1 Impact on System Reliability

We quantify the reliability P^i_{Reli} of an `aBeacon` device i with a percentage indicating among all the orders from a merchant deployed with the device i, how many orders we detect. In the real beacon systems, there are two major factors impacting the reliability, beacon device deployment environment, and couriers' smartphone hardware.

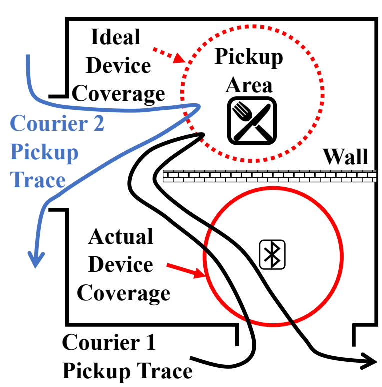


Figure 4.6: Impact of Environment

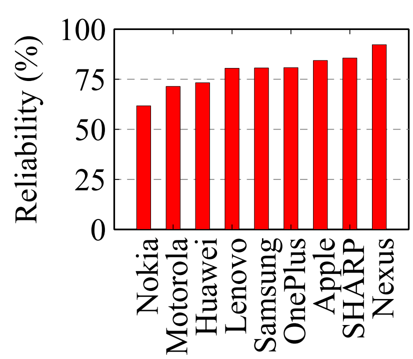


Figure 4.7: Impact of Smartphone

The deployment environment is an essential factor for reliability, as we found some merchants with an exceptionally low detection ratio. Although our deployment handbook suggested that "Beacons should be attached around the order pickup area", some business managers put devices somewhere else due to various reasons. For example, some merchants do not have a fixed "meal/groceries pickup area"; some merchants prefer the device to be placed somewhere else, e.g., under the counter. We performed some field study, and our findings can be clearly explained with Figure4.6 that depicts the layout of a real-world restaurant. In this merchant, there are two entrances with a horizontal wall in the center. Two couriers may pick up orders from both entrances, which leads to the different indoor pickup traces. Unfortunately, because the wall obstructs the device broadcast, only the Courier 1's arrival was detected, which results in a reliability of 46% in our observed period. If the `aBeacon` device were placed in the pickup area, we could have better reliability since both courier traces can be detected. In short, the impact of deployment position is difficult to estimate due to the uncontrollable deployment quality. The reason is we utilize our in-house business team with no deployment expertise (or incentive), and a deployed device can be moved as well, both of which typically leads to low reliability at some merchants.

The couriers' smartphone hardware is also an important impact factor. Our goal is to have most courier smartphones (if not all) to be compatible with `aBeacon` at both the hardware (i.e., phone brands and models) and software level (i.e., OS types). Given more than 109,000 couriers in Shanghai, it is challenging for us to either force the couriers to use specific smartphone brands or know if a courier uses an unsupported smartphone. To analyze the impact of smartphone OS, we divide all the orders in `aBeacon` merchants into two dimensions: whether its courier was detected by `aBeacon` or not; whether its courier was using an Android or iOS phone. Our data shows that, 63.4% of the total orders were detected with the Android couriers (including 52 brands and 672 models), and their average P^i_{Reli} is $\frac{63.4\%}{63.4\%+20.8\%} = 75.2\%$; 13.4% of the total orders were detected with the iOS couriers, and their average P^i_{Reli} is $\frac{13.4\%}{13.4\%+2.4\%} = 84.8\%$. We found iOS performs significantly better than Android. We have two conjectures for the reason: (1) iOS devices have better hardware (e.g., Bluetooth chips) since the hardware cost of some android phones are less than \$200, which is far less than that of iOS devices; (2) iOS has better resource scheduling strategies since some Android OS customize the Android strategy [121] and kill Bluetooth threads frequently to save energy. For different phone brands (different hardware), the average P^i_{Reli} varies. We show the average P^i_{Reli} of nine well-known brands in China in Figure4.7, in which Nexus has the highest P^i_{Reli} of 92%, and iPhones has a P^i_{Reli} of 84%.

4.3.2 Impact on Arrival Detection Model

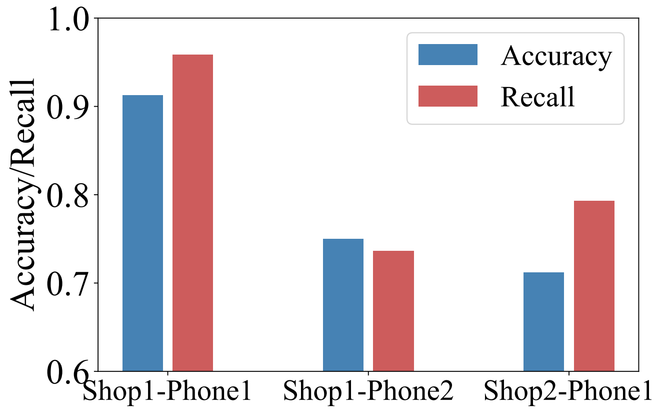


Figure 4.8: The Performance of the Traditional Status Estimation Model

To further reflect the impact of heterogeneities, we test the performance of the indoor arrival status estimation model based on CNN-LSTM [122] across different environments and different smartphone models. Data collected by phone1 in shop1 are used as the training dataset and three groups of data collected by phone1 in shop1, phone2 in shop1 and phone1 in shop2 are used as testing datasets. We use 5-fold cross-validation to split training and testing datasets, and compare the average accuracy and recall. Figure 4.8 shows that for the smartphone model that is unseen in the training process, the accuracy dropped from 91.25% to 75%, and the recall dropped from 95.83% to 73.61%. For the unseen shop, the accuracy of the status estimation model drops to 71.19%, and the recall drops to 79.27%. These results emphasize the significant influence of environment and smartphone model heterogeneities on the performance of the status estimation model.

4.4 Opportunities for Para-Pred

We find that large-scale gig delivery platform naturally provides interactive data which contains rich heterogeneities knowledge. There are usually a large number of couriers with different smartphone models to pick up orders in many different shops. So we obtain the Bluetooth interactive sensing data which contain the influence patterns of the heterogeneities.

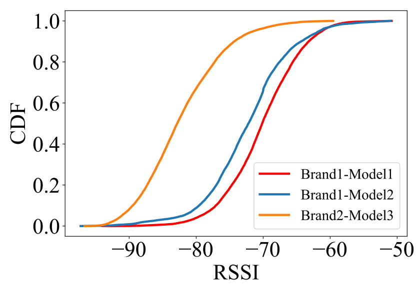


Figure 4.9: Similarity between the Influence Patterns of Smartphone Models

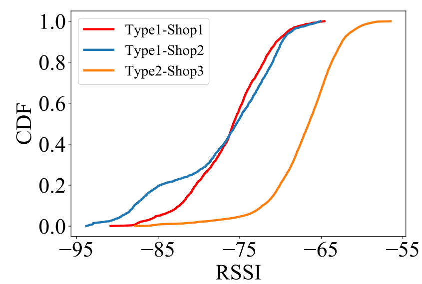


Figure 4.10: Similarity between the Influence Patterns of Shops

The different distributions of sensor data reflect the influence patterns of different environments and smartphone models on Bluetooth data. As shown in Figure 4.9, we analyze the distribution of Bluetooth signals detected by three different smartphone models in the same shop during the same courier's pickup time. The results show that data distributions of phone1 and phone2 of the same brand are more similar than that of phone3 of another brand. Similarly, we analyze the Bluetooth signal distribution of the same smartphone model in three different shops. As shown in Figure 4.10, we find that the data distributions of shop1 and shop2 of the same type (i.e., milk tea) are more similar than that of another type of shop3. It shows that there are similarities between the influence patterns of different smartphone models and between the influence patterns of different environments. Furthermore, for **target domains** for which we cannot train effective status estimation models due to the lack of data, we have the opportunity to use the similarity between influence patterns of **source domains** which have sufficient data to infer the status estimation model parameters of the target domain without extra data collection.

4.5 Related Works

4.5.1 Cross-domain Learning Methods

The research on how to improve the cross-domain generalization ability of the recognition framework fall into two categories: one is the method based on data conversion [109, 112, 111], and the other is the method based on adversarial learning [113, 114]. The main idea of the former is to learn how to transform the data features collected from one domain into the data features of other domains for the recognition model training. For example, CrossSense [111] learns a roaming model that transforms data features for each target domain and uses the generated data features for downstream tasks training.

The latter type uses the idea of adversarial learning to extract features irrelevant to the environment or user for recognition. For example, RFID [113] designs two domain discriminators to classify the user who performs this gesture or positions where the gesture is executed to extracts features irrelevant to the environments and users. However, when a new target domain appears, all these methods require re-collecting new data under limited conditions and model retraining, which are not efficient for large-scale scenarios. Our method infers the status estimation model for the new target domain according to the similarity between heterogeneities learned by the similarity extraction module without additional data collection and model retraining.

4.5.2 Graph Embedding based on GNN

The purpose of graph embedding is to map graph data into low dimensional dense vectors. It captures the topology of the graph, the relationship between vertices, and other related information. The graph neural network (GNN) model first appeared in [123], it extends the existing traditional neural network model and can be used to process data with arbitrary graph structure. Since GNNS effectively learn on graph structure data, there are several advanced graph embedding methods based on GNN. GraphSage [124] learns the nodes embeddings by aggregating information from neighbors in an inductive manner. Graph autoencoder (GAE) and variational graph autoencoder (VGAE) [125] use graph convolution network (GCN) encoder and simple inner product decoder to learn node embedding by minimizing the reconstruction error of adjacency matrix while considering graph structure and node feature information. [126] [127] further introduce the attention mechanism to learn the importance weights of neighbor nodes and the edge content. In our method, we incorporate the node and edge attributes and utilize a convolutional network and attention mechanisms to fully explore the features.

4.5.3 RSSI-based Sensing

Received Signal Strength Indication (RSSI) can be used to identify human activities [128, 129]. For example, in-air hand gestures around the user's mobile device can be recognized by analyzing the change of WiFi signal strength [128]. In recent years, the number of smartphones and devices equipped with BLE functions has increased. Therefore, many BLE sensing systems have been proposed. For example, [130] applies the idea of global map matching to route estimation based on BLE beacons. Given the known mapping of the BLE beacons and the couriers' smartphones, a straightforward solution is to mark the courier as 'arrival' when (s)he first scans the beacon.

4.6 `Para-Pred` Framework and Problem Definition

To address the heterogeneity problem in the beacon system for arrival detection, we design `Para-Pred`, an indoor status estimation framework based on the graph neural network, which directly **Pred**icts the effective indoor status estimation model **Para**meters for unseen scenarios. Our key idea is to utilize similarity between the influence patterns of heterogeneities on the Bluetooth signal to infer unseen scenarios' influence patterns directly. We evaluate the `Para-Pred` on 109,378 couriers with 672 smartphone models in 12,109 merchants from an on-demand delivery company. The evaluation results show that across environment and smartphone model heterogeneities, the accuracy and recall of our method achieve 93.62% and 95.20%, outperforming state-of-the-art solutions.

In this section, we introduce the framework and the problem definition of the design. In the following sections, we discuss the three components data processing module, similarity extraction module, and parameters prediction module in detail.

4.6.1 Framework

We design the model framework as shown in Figure 4.11, which consists of the following three modules.

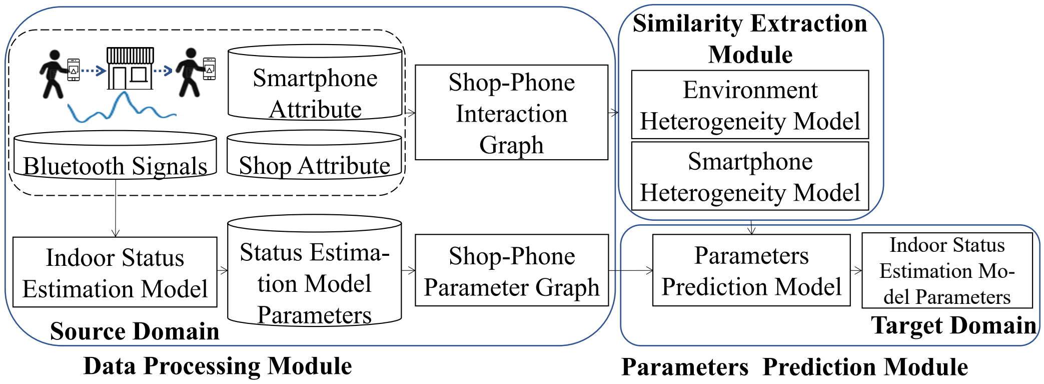


Figure 4.11: Framework of `Para-Pred`

Data Processing Module. This module prepares data for graph construction. First, original Bluetooth signals are processed into a suitable form as the edge attribute of the shop-phone interaction graph. Then we train the status estimation models for source domains with labeled data, and the model parameters are taken as the edge attribute of the shop-phone parameter graph.

Similarity Extraction Module. The purpose of this module is to learn the similarity between influence patterns of environments and between influence patterns of smartphone models on Bluetooth signals. First, we build a shop-phone interaction graph G to learn latent factors of smartphone model nodes and shop nodes that contain the influence patterns of heterogeneities through the environment heterogeneity model and smartphone heterogeneity model. Then we construct the shop-shop similarity graph and phone-phone similarity graph according to the similarity between smartphone model latent factors and between shop latent factors.

Parameters Prediction Module. We first learn effective indoor status estimation models for source domains and construct the shop-phone parameters graph based on the indoor status estimation model parameters T. Then we inherently combine T and similarity graphs G^s and G^p to predict the status estimation model parameters for the target domains.

4.6.2 Problem Definition

Definition1: Indoor Status Estimation Model Parameters. We utilize the CNN-LSTM as the indoor status estimation model to estimate the probability of arrival and departure at every time clip, which consists of one-layer CNN, one-layer LSTM, and a fully connected layer. The input is the RSSI time series, and the output is the predicted probability of the arrival or departure status. For the source domain with sufficient labeled data, we obtain the status estimation model parameters by training.

Definition2: Shop-Phone Interaction Graph. We define a shop-phone interaction graph $G = (U, V, E)$, to learn the latent factors representing the influence patterns for each shop and smartphone model. U and V are the sets of shops and smartphone models respectively. $E \subseteq U \times V$ is the set of interactive edges, and the edge $E(u_i, v_j)$ denotes there are history detection records between shop u_i and smartphone model v_j.

Definition3: Similarity Graph. We define that $G^s = (U, E^s)$ and $G^p = (V, E^p)$ are the shop-shop similarity graph and the phone-phone similarity graph, respectively, where $E^s(u_i, u_j) = 1$ if the similarity of the latent factors of u_i and u_j which obtained in shop-phone interaction graph is greater than the threshold ϵ, similarly for $E^p(v_i, v_j) = 1$

Definition4: Shop-Phone Parameters Graph. We utilize shop-phone parameters graph $T = (U, V, E^t)$ to predict status estimation model parameters for target domains, where $E^t \subseteq U \times V$ represents the set of parameters edges, and the edge $E^t(u_i, v_j)$ denotes there are status estimation model parameters of the source domain (i.e. smartphone model v_j in shop u_i) obtained by training.

We aim to utilize (1) the similarity between the influence patterns of environments and (2) the similarity between the influence patterns of smartphone models to predict the effective indoor status estimation model parameters for target domains. Then the status estimation models composed of prediction parameters can be directly used for status estimation in unseen scenarios. Formally, the problem is defined as: $\hat{e}_{ij}^t = F_\theta(G, T)$. We input interaction graph G and parameters graph T to learn a model F_θ that can predict the status estimation model parameters $\hat{e}_{ij}^t$ for unseen scenarios (e.g., smartphone model v_j in shop u_i). Then we input the RSSI time series r_{ij} detected by smartphone model v_j in shop u_i to the status estimation model (i.e., parameters $\hat{e}_{ij}^t$) to predict status probability of r_{ij} described as $y' = ConvLSTM(r_{ij}; \hat{e}_{ij}^t)$. Training is a one-off cost performed off-line, and the learned model parameters $\hat{e}_{ij}^t$ can be used for unseen scenarios without incurring extra training.

4.7 Design of `Para-Pred`

In this section, we detail each model component and describe the design of loss function and the training method.

4.7.1 Data Processing Module

First, due to the packet loss in the process of data collection, we interpolate the RSSI time series and utilize the Kalman filter [131] to remove the abnormal values. To process the data into the form required by the shop-phone interaction graph G, we utilize DTW to align all RSSI time-series detected by the same smartphone model in the same shop during pickup time. Then, we calculate the mean vector $e_{ij} = \frac{1}{L}\sum_{l=1}^{L} r_l$ as the attribute of E, where r_l is the lth historical Bluetooth signal time-series detected by smartphone model v_j in shop u_i, L is the number of time series.

To obtain the indoor status estimation model parameters of source domains for shop-phone parameters graph construction, we divide the preprocessed time series into a set of sliding windows whose length is m (12 in `Para-Pred`) with 50% overlap and input them to the indoor status estimation model. After training, we obtain the status estimation model parameters e_{ij}^t for smartphone model v_j and shop u_i as the attribute of E^t of the shop-phone parameters graph. In addition, the shop attribute dataset includes the information of shop latitude, longitude, and main category, which capture the common attribution of shops. And the smartphone model attribute dataset includes the smartphone model and smartphone OS types.

4.7.2 Similarity Extraction Module

The similarity extraction module aims to learn the influence patterns of heterogeneities from historical interaction data and obtain the similarity between the influence patterns of environments and between the influence patterns of smartphone models. We define the shop-phone interaction graph G and utilize the environment heterogeneity model and smartphone heterogeneity model to learn latent factors of shop and smartphone model nodes that contain their influence patterns and output the shop-shop similarity graph G^s and phone-phone similarity graph G^p as shown in Figure 4.12. Next, we introduce the environment heterogeneity model, phone heterogeneity model, and how

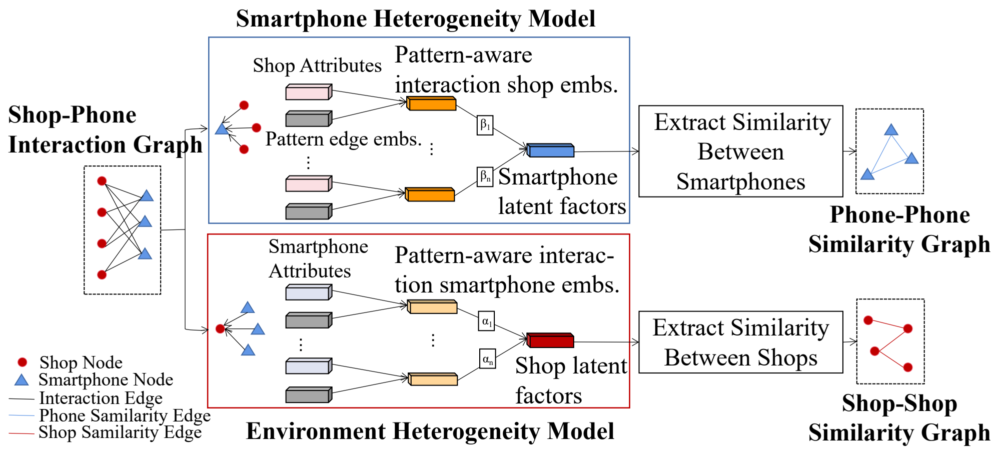


Figure 4.12: Structure of Similarity Extraction Module

to obtain the similarity graphs in detail.

Environment Heterogeneity Model

This model aims to learn shop latent factors which are utilized to model the influence patterns of shops, denoted as $h_i \in R^d$ for shop u_i. The edge attribute $e_{ij} \in R^{1\times f}$ of shop-phone interaction graph G are generated from the historical Bluetooth signal time series of smartphone model v_j in shop u_i, which contains the influence patterns of environment heterogeneity. Graph structure also contains rich interactive information between shops and smartphone models. So we jointly capture interactions and historical sensing data to further learn shop latent factors h_i.

To consider the different contributions of neighbor smartphone model nodes to the shop node in the process of aggregating information, we introduce the attention mechanism based on edge and node attributes. The contribution of the smartphone neighbor v_x to the shop u_i is computed as follows:

$$\alpha^*_{ix} = \sigma(W \cdot [\{g_{ix}, \forall x \in P(i)\} \parallel s_i] + b) \tag{4.1}$$

where W and b are the weight and bias of a neural network, $\parallel$ denotes the concatenate operation, σ denotes the activation function, s_i is the attribute of shop u_i, $P(i)$ represents the set of neighbor smartphone model nodes interacted with shop u_i, g_{ix} is the pattern-aware interaction embedding of smartphone model v_x which jointly capture the interaction information and influence pattern.

To model the g_{ix}, firstly, we introduce a pattern edge embedding m_{ix} to model the influence pattern of shop u_i by inputting the edge attribute e_{ix} to a 1D convolutional neural network (1D-CNN) ϕ_m, i.e., $m_{ix} = \phi_m(e_{ix})$. Then, we feed the combination of the m_{ix} and the smartphone model v_x's attribute p_x into a Multi-Layer Perceptron (MLP) ϕ_v: $g_{ix} = \phi_v([m_{ix}||p_x])$. To make the contribution coefficients easy to compare among different neighbors of shop u_i, we utilize softmax function to standardize them as $\alpha_{ix} = \frac{exp(\alpha^*_{ix})}{\sum_{x \in P(i)} exp(\alpha^*_{ix})}$. Then, the shop u_i's latent factor h_i is aggregated by the smartphone neighbor's pattern-aware interaction embeddings and contribution coefficients, which is defined as:

$$h_i = \sigma(W \cdot \left\{ \sum_{x \in P(i)} \alpha_{ix} g_{ix} \right\} + b) \tag{4.2}$$

Smartphone Heterogeneity Model

This model aims to learn smartphone model latent factors $z_j \in R^d$ which reflect the influence patterns of smartphone model v_j. Shop-phone interaction graph G includes abundant interactive sensing data which contains the influence pattern of smartphone models on Bluetooth signals from different shop perspectives. Therefore, interactive information and sensing data should be considered together to learn smartphone model latent factors.

Similarly, we aggregate the pattern-aware interaction embedding of shops interacting with smartphone model v_j to learn the smartphone model latent factor z_j based on attention mechanisms. To mathematically represent the aggregation process, we utilize the following function: $z_j = \sigma(W \cdot \left\{ \sum_{o \in S(j)} \beta_{jo} a_{jo} \right\} + b)$, where $S(j)$ is a set of neighbor shop nodes interacted with smartphone model v_j, a_{jo} is a pattern-aware interaction shop embedding which is obtained by inputting the pattern embedding m_{jo} and shop u_o's attribute s_o into a MLP ϕ_u, i.e. $a_{jo} = \phi_u([m_{jo}||s_o])$.

We consider the pattern-aware interaction shop embedding and the smartphone model v_j's attribute p_j to learn the contribution coefficient β_{jo} for neighbor shop u_o, as below, $\beta^*_{jo} = \sigma(W \cdot [\{a_{jo}, \forall o \in S(j)\}||p_j] + b)$, $\beta_{jo} = \frac{exp(\beta_{jo}{}^*)}{\sum_{o \in S(j)} exp(\beta^*_{jo})}$.

Obtain the latent factors of shops and smartphone models

To learn the latent factors of shops and smartphone models, which contain the influence patterns of heterogeneities from shop-phone interaction graph G, we first concatenate the latent factors of shop u_i and phone v_j (i.e., h_i and z_j) and then put them into MLP to reconstruct edge attributes, i.e., $\hat{e}_{ij}^{(k)} = \sigma(W^{(k)} \cdot \hat{e}_{ij}^{(k-1)} + b^{(k)})$, where $\hat{e}_{ij}^{(0)} = [h_i||z_j]$, k is the index of a hidden layer. We consider the output of the last layer as the reconstruct edge attributes, i.e., $\hat{e}_{ij} = \hat{e}_{ij}^{(k)}$. Then we construct the following loss function:

$$L_r = \frac{1}{|Q|} \sum_{i,j \in Q} \|\hat{e}_{ij} - e_{ij}\|_2 \tag{4.3}$$

where $|Q|$ is the number of observed interactive edges and e_{ij} is the ground truth edges in the shop-phone interaction graph.

Extract Similarity

We aim to construct similarity graphs based on the similarity. We measure the similarity by calculating the euclidean distance between latent factors of shops and between latent factors of smartphone models, respectively, i.e. $dis_{ij}^h = ||h_i, h_j||_2$ and $dis_{ij}^z = ||z_i, z_j||_2$. And the shop-shop similarity graph G^s is generated according to the dis_{ij}^h. If $dis_{ij}^h <= \epsilon$, there is a similarity edge between shop u_i and shop u_j, where ϵ is the distance threshold that controls the sparsity of the shop-shop similarity graph. The phone-phone similarity graph G^p is obtained in the same way.

4.7.3 Parameters Prediction Module

The parameters prediction module infers the parameters of indoor status estimation models for target domains based on (1) the similarity between shops'influence patterns and (2) the similarity between smartphone models'influence patterns, respectively. As shown in Figure 4.13, the input of this module is a shop-phone heterogeneous graph composed of three sub-graphs: i.e., the shop-shop similarity graph G^s, phone-phone similarity graph G^p and incomplete shop-phone parameter graph T. The output is a complete shop-phone parameter graph, where we obtain the effective indoor status estimation model for the target domain according to the predicted status estimation model parameters.

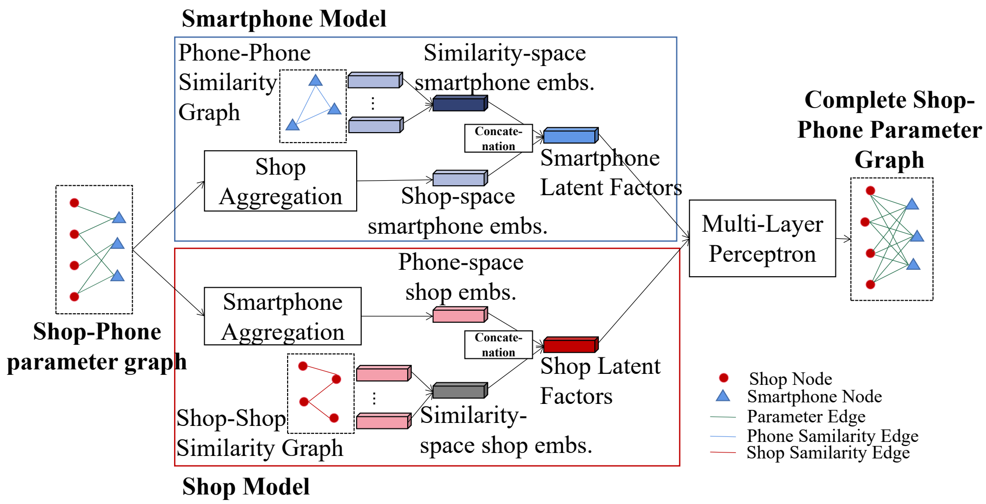


Figure 4.13: Structure of Parameters Prediction Module

Shop Model

To learn the latent factors $h_i^t \in R^d$ for shop u_i, we combine the shop-phone parameter graph and shop-shop similarity graph. The shop-shop similarity graph contains the similarity information between different shops' influence patterns which helps us learn the latent factors of shops from a more comprehensive perspective. Specifically, We first learn the phone-space shop embedding h_i^{t1} by smartphone aggregation from the shop-phone parameter graph T. Then we learn the similarity-space shop embedding h_i^{t2} from the shop-shop similarity graph G^s. Finally, we combine these two embeddings to obtain the final shop latent factors h_i^t.

To obtain the phone-space shop embedding h_i^{t1}, we aggregate the status estimation model parameter and interaction information from the set of smartphone models $P(i)$ which interact with shop u_i, to reflect the shop's influence pattern from the perspective of indoor status estimation model parameters. To mathematically represent this smartphone aggregation, we utilize the following function:

$$h_i^{t1} = \sigma(W \cdot \left\{ \sum_{a \in P(i)} \gamma_{ia} q_{ia} \right\} + b) \tag{4.4}$$

where q_{ia} is the parameter-aware smartphone model embedding which models the interaction and model parameter information between shop u_i and smartphone model v_a. To obtain the q_{ia}, we first introduce a parameter embedding n_{ia} to model parameters between u_i and v_a by inputting the edge attribute e^t_{ia} to a MLP φ_p, i.e., $n_{ia} = \varphi_p(e^t_{ia})$. Then we feed the concatenation of n_{ia} and the smartphone model v_a's attribute p_a into a MLP φ_v. And q_{ia} is defined as $q_{ia} = \varphi_v([n_{ia}||p_a])$. We design a **smartphone attention** γ_{ia} to represent the importance of the interaction with smartphone model v_a in contributing to phone-space shop embedding h^{t1}_i. We input the parameter-aware smartphone model embedding q_{ia} and shop u_i's attribute s_i into a one-layer neural network to obtain the smartphone attention γ_{ia}, i.e., $\gamma_{ia} = \frac{exp(\sigma(W \cdot [q_{ia}||s_i]+b))}{\sum_{a \in P(i)} exp(\sigma(W \cdot [q_{ia}||s_i]+b))}$.

Since the influence patterns of shops $C(i)$ which are similar to the shop u_i's help us model the parameter information of the shop u_i, we capture the similarity information to learn shop latent factors. We aggregate the phone-space shop embedding h^{t1}_c of u_i's similarity neighbor shops $C(i)$ to learn the similarity-space shop embedding h^{t2}_i, as the follows:

$$h^{t2}_i = \sigma(W \cdot \left\{ \sum_{c \in C(i)} \mu_{ic} h^{t1}_c \right\} + b) \tag{4.5}$$

We model the contribution strengths of similarity neighbor shop nodes by relating the **shop similar neighbor attention** μ_{ic} with h^{t1}_c and shop attribute s_i, as $\mu_{ic} = \frac{exp(\sigma(W \cdot [h^{t1}_c||s_i]+b))}{\sum_{c \in C(i)} exp(\sigma(W \cdot [h^{t1}_c||s_i]+b))}$.

We consider the shop-phone parameter graph and shop-shop similarity graph together to obtain the final shop latent factor h^t_i because both graphs provide model parameter information for shop u_i from different aspects. We input the h^{t1}_i and h^{t2}_i to MLP to obtain the final shop latent factor h^t_i, $h^{t(k)}_i = \sigma(W^{(k)} \cdot h^{t(k-1)}_i + b^{(k)})$ where $h^{t(0)}_i = [h^{t1}_i||h^{t2}_i]$.

Smartphone Model

Similar to the shop model, we first learn the shop-space smartphone model embedding z^{t1}_j by aggregating the interaction and model parameter information from shops $S(j)$ which interact with smartphone model v_j from shop-phone parameter graph T based on the **shop attention** ρ. Then we obtain the similarity-space smartphone model embedding z^{t2}_j by aggregating the shop-space embeddings of similarity neighbor smartphone models $N(j)$ that are similar to v_j from phone-phone similarity graph G^p based on the **smartphone similar neighbor attention** τ. Finally z_j^{t1} and z_j^{t2} are jointly fed into MLP to obtain the final smartphone model latent factors z_j^t.

Predict Model Parameters.

We utilize the shop latent factor h_i^t and the smartphone model latent factor z_j^t to learn effective model parameters for target domains. We feed the concatenation of them into MLP to predict missing model parameters $\hat{e}_{ij}^t$ as follows:

$$\hat{e}_{ij}^{t(k)} = \sigma(W^{(k)} \cdot e_{ij}^{t(k-1)} + b^{(k)}) \tag{4.6}$$

where $e_{ij}^{t(0)} = [h_i^t || z_j^t]$.

Then, to trade off the missing parameter edges prediction task in shop-phone parameter graph T and the status estimation task, we construct the following two common loss functions:

$$L_p = \frac{1}{|O|} \sum_{i,j \in O} \left\| \hat{e}_{ij}^t - e_{ij}^t \right\|_2 \tag{4.7}$$

where $|O|$ is the number of unknown parameter edges between shops and smartphone models in the training set, $\hat{e}_{ij}^t$and e_{ij}^t are the predicted and ground truth indoor status estimation model parameters between these shops and smartphone models. We utilize this loss function for the missing parameter edges prediction task.

For the status estimation task, we input the RSSI time series r_{ij} detected by smartphone model v_j in shop u_i to the status estimation model CNN-LSTM consists of $\hat{e}_{ij}^t$ parameters. So, the predicted status probability of r_{ij} can be described as $y' = ConvLSTM(r_{ij}; \hat{e}_{ij}^t)$. Thus, the loss function for this task is obtained by:

$$L_s = \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} -y_{im} log y'_{im} \tag{4.8}$$

where y_i is the ground truth status, N is the number of series, M is the number of status.

Since our ultimate goal is to improve the performance of status estimation, the overall loss function L is derived as: $L = L_p + \eta L_s$, where η is a trade-off parameter.

4.7.4 Model Training

To obtain the status estimation model parameters of each source domain for shop-phone parameter graph construction and reduce the training cost, we first utilize all labeled data from source domains to train a unified status estimation model. Then we utilize the data of each source domain to fine-tune the CNN layer of the unified model to obtain the status estimation model parameters for each source domain. To obtain the similarity graphs, we optimize the loss function L_r to learn the shops' and smartphone models' latent factors of the shop-phone interaction graph. Finally, to predict the missing parameter edges that are effective for the status estimation task, we optimize the loss function L to jointly train the parameter edge prediction task and status estimation task. We adopt Adam as the optimizer in our training process. To prevent overfitting, we apply the dropout strategy to our model.

4.8 Evaluation of `Para-Pred`

4.8.1 Experiment Setup

Data Collected

`aBeacon` **Data:** As shown in Table 4.1, the aBeacon dataset includes 64 million delivery orders involved with 109,378 couriers with 672 smartphone models in 12,109 merchants in Shanghai. The period of this dataset is from 2019/07/01 to 2020/10/31. We show the evaluation results in a sub-region of Shanghai (i.e., 6km $\times$ 6km) including 269 active couriers with 143 smartphone models, 187 active merchants, and around 4341 delivery orders on each day for four months.

Table 4.1: `aBeacon` Data

Attribute	Example	Attribute	Example
Courier ID	C_000001	RSSI	-80dB
Timestamp	2019/08/15 12:30:23	Phone ID	D_000001
Device ID Tuple	(UUID, Major, Minor)	Phone Brand/OS	Apple/iOS
Merchant ID	M_000001	Phone Model	iPhone X

Table 4.2: Courier Report Data

Attribute	Example
Courier ID	C_000001
Timestamp	2019/08/15 12:30:23
Merchant ID	M_000001
Order ID	O_000001
Report Type	Acceptance/Arrival/ Departure/Delivery

Courier Report Data: As shown in Table 4.2, for each delivery order, the report data record the time of four major status, i.e., accepting an order, arrival at the merchant, departure from the merchant (with the order), and final delivery to the customer. We obtain the status of couriers because these timestamps are uploaded by courier smartphone apps.

Parameter Settings

We implement our method and baselines with Pytorch 1.3.1 in Python 3.7.5 environment and train these with 16 GB memory and Tesla V100-SXM2 GPU. The four-month aBeacon monitoring data are divided into four months, we utilize the first three months' data to train our model and the last month's data as the evaluation.

Baselines

In order to prove the superiority of our method, we compare it with following classic and state-of-the-art methods.

• SVM [132]: This model utilizes the support vector machine as the classifier, we extract the time-domain features including mean, standard deviation, skewness, kurtosis, max, min, shape factor and the frequency-domain features including FFT Peaks, energy, and domain-frequency.

• C-LSTM [122]: This method discovers deep features from time series for status estimation without extracting features manually.

• CrossSense [111]: This method proposes a roaming model based on machine learning to transform the signal features from source domains to target domains and adopts multiple expert models for recognition.

• EUI [113]: This method contains two domain discriminators to classify the user or the position of the gesture with a multi-task optimization in the environment and user invariant training via adversarial learning. In our problem, we convert to design the environment and smartphone model domain discriminators.

• GraphSage [124]: This is a graph method, which transfers information through the edge without using the attributes of nodes and edges to attend neighbor nodes. We use this method to learn the shop-phone parameter graph for the status estimation model parameter prediction.

4.8.2 Resulta and Analysis

Overall performance

We compare our work with the above baselines, and the average results of accuracy, recall, precision, and F1-score of arrival and departure status are shown in Table 4.3. We observe that our method has the best overall performance compared with other methods. In addition, our method does not need extra data re-collection and model retraining when facing new target domains. `Para-Pred` achieves more than 17% higher precision and F1-score than the classic baselines(i.e., SVM and C-LSTM), because we consider the impact of the heterogeneities. The average accuracy and recall of `Para-Pred` are 93.62%, and 95.20%, respectively, which advantage the EUI by 11.87% and 14.54%. Then, the advantage of our `Para-Pred` over GraphSage validates the effectiveness of our aggregation method and attention mechanisms. We also conduct a t-test to show that our results are statistically significant with the p-value < 0.01 compared to the best baseline EUI.

Table 4.3: Overall performance

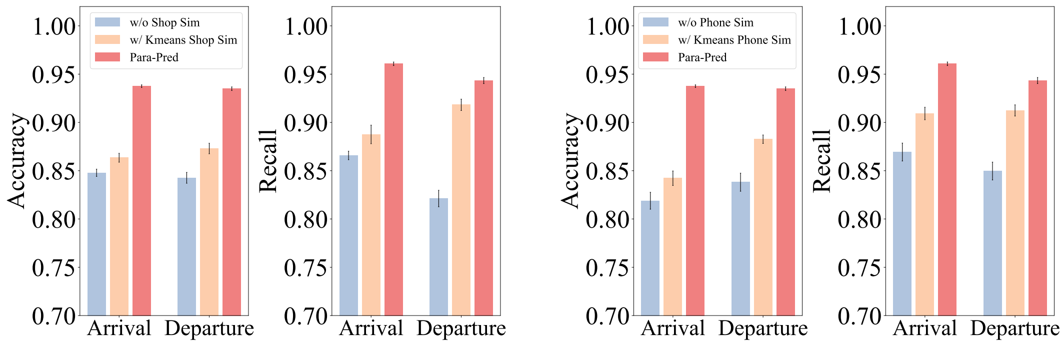
Model	**SVM**	**C-LSTM**	**CrossSense**	**EUI**	**GraphSage**	**Our Model**
Accuracy	65.56%	72.82%	77.93%	81.75%	80.88%	93.62%*
Recall	62.79%	71.46%	79.93%	80.66%	82.22%	95.20%*
Precision	63.56%	73.64%	76.86%	80.94%	77.71%	91.27%*
F1-score	64.88%	72.11%	75.71%	79.90%	77.66%	92.17%*

* the result is significant according to T-test at level 0.01 compared to EUI.

Significance of Similarity extraction module

Here we analyzes the importance of the similarity learned by the similarity extraction module for system performance.

Impact of Environment Heterogeneity Model To illustrate the importance of the similarity between environmental influence patterns learned by the environment heterogeneity model for our model performance, we compare our model with two variants. They are defined as: **(1) w/o Shop Sim:** We remove the shop-shop similarity graph from the parameter prediction module. In this variant, we simply learn the shop latent factors h_i^p from the shop-phone parameter graph. **(2) w/ Kmeans Shop Sim:** We adopt Kmeans, a traditional clustering method, to cluster the raw sensor data and



(a) (b)

Figure 4.14: Significance of Similarity Extraction Module.

connect shops in the same class to construct a new simple shop-shop similarity graph instead of the shop-shop similarity graph learned by our similarity extraction module.

The comparison results are shown in Figure 4.14a, the average accuracy and recall of arrival and departure status of our method are higher than 93.0% and 94.3% severally, which is better than these two variants. The w/o Shop Sim method performing worst proves the importance of the similarity information between influence patterns of shops. Our `Para-Pred` performing better than the w/ Kmeans Shop Sim method proves the advantage of our environment heterogeneity model which learns influence patterns of environments and obtains the effective similarity information by considering the interaction information contained in sensor data.

Impact of Smartphone Heterogeneity Model Similarly, to intuitively show how the similarity between smartphone influence patterns learned by the smartphone model heterogeneity model affects our model performance, we evaluate our model and two variants, **w/o Phone Sim** and **w/ Kmeans Phone Sim**. Figure 4.14b shows that our `Para-Pred` performs best, it proves that the smartphone heterogeneity model extracts effective smartphone model influence patterns and the similarity information is important to improve the performance of final status estimation.

To sum up, the similarity extraction module learns the influence patterns of heterogeneities and the similarity information effectively and improves the overall performance of our system.

Impact of Environment Diversity.

The purpose of this section is to study the impact of different environment types on the performance of `Para-Pred`. Taking the arrival estimation as an example, we select the samples detected by the same smartphone in five different types (fried chicken, breakfast, milk tea, light meal, and snack bar) of shops for testing. These five types are common in daily life and have different environmental characteristics. As shown in Figure 4.15, the performance of different shop types is slightly different, the average accuracy is more than 91% and the average recall is more than 89% in all types of shops. In general, our framework is robust for different environment types.

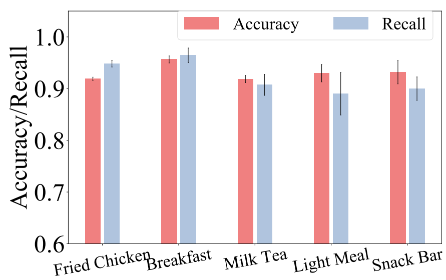


Figure 4.15: Performance of Different Environments.

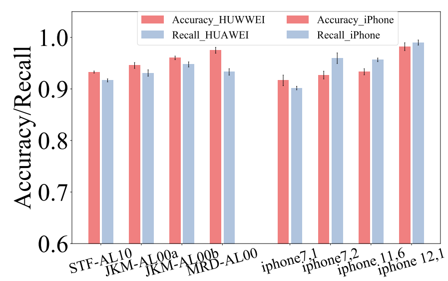


Figure 4.16: Performance of Different Smartphone Models

Impact of Smartphone Diversity.

To verify the performance of `Para-Pred` in different smartphone models, we test our model with the data detected by eight different smartphone models under two well-known brands (HUAWEI and iPhone) in the same shop for arrival estimation. The abscissa in Figure 4.16 is sorted by the release time of smartphones from the same brand, and the results show that iPhone 12,1 has the highest average accuracy and recall of 98.20% and 99.09%, respectively. We find for smartphone models from the same brand, the newer the smartphone, the better the average performance. In a nutshell, our model performs well for different smartphone models.

4.9 Discussion

4.9.1 Lesson Learned

(1) We analyze the influence patterns of the environment and smartphone heterogeneities and find that there is similarity information in them. (2) The similarity information is important for predicting effective indoor status estimation models for target domains. (3) Graph learning works well for addressing the heterogeneities in large-scale indoor status estimation since the influence patterns of heterogeneities and the similarity information can be effectively learned and represented. (4) The evaluation results show that our method alleviates the heterogeneities effectively for indoor status estimation in large-scale scenarios. We also find that our method is robust for different types of environments, and the release time of smartphone models from the same brand has a positive impact on system performance.

4.9.2 Generalizability

Although `Para-Pred` is designed for on-demand delivery, we believe the underlying ideas of addressing the impact of heterogeneities on sensing data can be generalized to other BLE scenarios, such as interaction in museums [133], airports [134], and TraceTogether [135]. For example, in airports [134], passengers receive tailored information as they arrive at the lounge or retail areas. In this scenario, Bluetooth signals detected by different smartphones are input to the similarity extraction module to learn the similarity between the effects of heterogeneities. The parameter prediction module predicts the status estimation model parameters for new scenarios. We believe `Para-Pred` with some modifications is a potential solution to mitigate the heterogeneity effects in these application scenarios, which provides more accurate status estimation and better services to users.

Chapter 5

Travel Time Prediction with Graph Learning based on Encounters

5.1 Introdution

Based on the data collected from the sensing layer (e.g., arrival events detected by beacons), we can do many prediction works to infer a variable that is either unobservable directly through sensing (e.g., indoor locations in some scenarios) or observable only if it happens in the future (e.g., estimated time of arrival) with the help of advanced machine learning and data mining techniques. In this chapter, we will dive deep into an indoor localization problem where we transfer it into a travel time estimation and address it based on the encounter sensing and graph learning.

Nowadays, on-demand delivery [136, 137, 1, 87] is an emerging business for Gig Economy [79] where gig workers deliver orders (e.g., food) within a short time (e.g., 30 minutes) from merchants to customers. This business grows rapidly with several on-demand delivery platforms worldwide (e.g., DoorDash [83] and Eleme [1]). To achieve timely delivery, couriers' real-time localization is one of the indispensable supporting services involving all the stakeholders including couriers, merchants, customers, and platforms such as courier navigation [138], merchants' order preparation [139], status tracking for customers [140], and platform order dispatching [87].

To achieve timely delivery, couriers' real-time localization is one of the indispensable supporting services involving all the stakeholders including couriers, merchants, customers, and platforms such as courier navigation [138], merchants' order preparation [139], status tracking for customers [140], and platform order dispatching [87]. While outdoor locations can be obtained by smartphone GPS accurately [141], indoor locations are difficult to acquire due to the weak GPS signal. Given many merchants' shops are located in multi-story malls in urban areas (e.g., 9,562 shops located in 576 malls in Shanghai City), couriers' indoor localization becomes the bottleneck of improving the user experience and operational efficiency in on-demand delivery.

The state-of-art indoor localization solutions can be organized using the following two-dimensional taxonomy:

• *Absolute* Versus *Relative*. Absolute localization refers to localization in a single predetermined coordinate system (e.g., GPS) or map (e.g., floor plan) with concrete coordinates. Relative localization refers to localization in the context of one's neighbors or local environment [142, 143], usually with additional ranging or odometry information.

• *Infrastructure-based* Versus *Infrastructure-free*. In infrastructure-based solutions, some infrastructures such as Wi-Fi APs [91, 144, 145, 146, 147, 148], LED fixtures [149, 94], RFID tags [150], and PIR sensors [151, 30] are used as *anchors infrastructures* to localize the nearby *target* devices but introduce a high cost for deployment in multiple environments. Particularly, a Bluetooth beacon system, aBeacon [1], was built by Alibaba, which cost more than $100K and retired within two years. In infrastructure-free solutions, landmarks such as acoustic [152, 153], light [154], magnetic [155], and electromagnetic [156] are mapped to fixed locations on floor plans. These methods usually introduce a high cost of maintaining a fingerprints database.

Given the practical limitations, absolute localization is inapplicable due to no accurate GPS or sufficient floor plans, and infrastructure-based methods are inapplicable due to the high cost across multiple environments. Admittedly, relative infrastructure-free localization has been studied (e.g., TransLoc [136]), where they only use couriers' reporting at merchants to obtain anchor information. However, the sparsity and uncertainty of couriers' reporting behavior lead to unsatisfactory performance in accuracy and robustness, severely restricting the upper-layer applications.

We explore *couriers' indoor encounters* as an opportunity to advance the state-of-practice. The encounters can play a key role in couriers' indoor localization for two reasons: (i) frequent indoor encounters convey comprehensive spatial-temporal information that can be aggregated and shared between present and subsequent encounters; (ii) encounter events among couriers can be detected at low cost via Bluetooth Low Energy (BLE) advertising and scanning on couriers' smartphones under the couriers' consent.

Admittedly, although utilizing encounter information for localization is not new, existing solutions [157, 158, 159, 160, 88] are not applicable due to the following practical challenges in on-demand delivery. **(i) Lack of odometry information.** Most solutions rely on IMU-based dead reckoning to measure the relative distance to anchor infrastructure. However, IMU data on couriers' phones might be difficult to use due to accuracy (calibration lacking for 672 smartphone models of 52 brands used) and privacy (gait may leak identity information [161, 162]) issues. **(ii) Limited anchor information.** Even for encounter-based solutions, a certain amount of anchor infrastructure or semantic anchors with known locations are needed to provide initial location information. However, utilizing infrastructures limits the scale-up ability of the solution. Couriers' reporting events at merchants are used as semantic anchors in TransLoc [136], but the reporting event is sparse (two per delivery order), limiting the performance.

To tackle the challenges, we design, prototype, and evaluate `P`^{2}`-Loc`, a localization system based on **p**eer to **p**eer encounters that only use couriers' accounting data and encounter data to infer their relative locations. Specifically, we build a graph neural network (GNN) [163, 164] and utilize the idea of link prediction [165] to infer the couriers' relative locations (i.e., travel time) to all the indoor merchants from sparse reporting events and massive encounter events. In doing so, we build on recent progress in graph-based deep learning to solve the domain-specific problem in a data-driven fashion. Particularly, historical encounter and travel time data are used as labels hence no additional efforts are needed to collect labels. The contributions of this work are three-fold.

• To the best of our knowledge, we are the first to use encounters to build, prototype, deploy, and evaluate a relative, infrastructure-free indoor localization system in a real-world application, i.e., on-demand delivery. Based on the ubiquitous encounter events and couriers' indoor mobility preferences, we infer the couriers' real-time locations without extra infrastructure costs, making our solution scalable for a large-scale commercial setting. We also show that the idea of utilizing encounter information for relative localization can be generalized to other problems. We share the source code [?] and one month of the data we collected in P^2-Loc [166] for the research community to validate our results and conduct further research.

• We build a GNN model to aggregate the information in couriers' encounters and infer their real-time locations. To tackle the odometry lacking, we build a graph to implicitly learn the topology of indoor merchants from couriers' historical indoor travel data; to tackle the anchor information lacking, we use GNN to integrate node information and the topological structure of the graph and use link prediction to predict couriers' travel time to all the merchants. Unlike GNNs in recommendation systems where the graph is static with binary edge features, the graph in P^2-Loc is temporally correlated with multiple features on heterogeneous edges. We design an embedding network to embed the edge information to the same space with nodes and a recurrent module to utilize short-term memory.

• We prototype and implement P^2-Loc on a commercial on-demand delivery platform, and evaluate P^2-Loc in a mall with 4,075 couriers and 79 merchants for a month. The results show that P^2-Loc outperforms methods based on Wi-Fi, GPS and reporting by 9%, 19%, and 51%, respectively, and outperforms other encounter-based methods (i.e., MDS-based and statistical) by 8% and 31%. As a concrete application of P^2-Loc, we show that the same delivery order scheduling algorithm with better localization results from P^2-Loc can reduce the platform's overdue rate to save \$ 40,000 every day via an offline analysis of real-world data. The evaluation and application results lead to some key lessons learned on the trade-off between the performance and model complexity.

5.2 Related Works

5.2.1 Relative Encounter-based Localization

Relative localization and encounter-based localization are closely related because relative localization usually relies on communication among neighboring targets. The related works can be categorized into *target-relative* and *anchor-relative*.

• In target-relative works, the targets only need to decide their relative locations to other targets. Graph realization methods (e.g., MDS) based on ranging are used to calculate the relative distance directly [167, 168], or correct the dead reckoning results [169]. EASE [157] proposes a distributed algorithm to infer distance among all nodes in the setting that each node knows its own location and encounter information with other nodes. The idea of relative localization is also adopted in the Internet system to assign coordinates to the hosts given the round trip time [170]. Bluetooth and Wi-Fi data from smartphones are used in [143] to decide the relative locations of vehicles. Spatial-temporal phase information is used in [142] to decide the relative locations of RFID tags.

• In anchor-relative works, the targets need to decide their relative locations to some anchors with the help of nearby targets. A hybrid solution is proposed in [171] to select some anchor nodes and use them to localize the others. Encounter information from acoustic sensing is used in [159] to localize and navigate users in the indoor environment with pre-placed beacons, and in [160] to refine Wi-Fi localization results. GPS errors are modeled and eliminated in [172] based on the raw GPS data from nearby smartphones. Social-Loc [88] uses the encounter information between smartphones to improve the Wi-Fi-based localization. CoSMiC [173] uses the encounters information from Wi-Fi to recover the trace of lost children. Encounter information from Bluetooth is used in [174] to localize devices relative to Wi-Fi APs.

Compared with the existing relative encounter-based works, the contribution of our works is that we use graph learning techniques to aggregate the heterogeneous spatial-temporal information from couriers' encounters, and build an infrastructure-free, and odometry-free localization system that works on off-the-shelf smartphones in a sparse-anchor environment.

5.2.2 Deep Learning for Mobile Applications

Deep learning is becoming increasingly important for mobile applications with the availability of large-scale data from mobile devices [175, 176]. Early works mostly focus on city-wide applications such as traffic forecasting [177, 54], bike mobility modeling [178], and ride-hailing [61], while recent works start to focus on finer-grained applications such

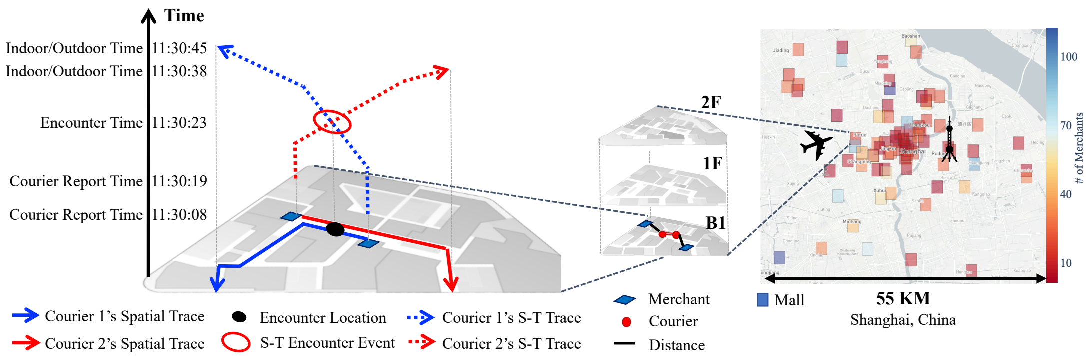


Figure 5.1: Illustration of Two Courier's Encounter. (1) Two couriers encountered on their way out of the mall after picking up the order on the B1 floor. The couriers' relative locations to the two merchants can be measured as the travel time between their departure and the encounter based on the shortest-path observation. (2) The right map shows all the 116 malls in Shanghai.

as gesture recognition [27], health monitoring [31], and indoor localization [179, 148].

5.3 Opportunities

5.3.1 ETA for Relative Localization

Unlike applications that need targets' *absolute* locations on a predefined map, on-demand delivery only needs targets' *relative* locations to the indoor merchants, hence offering a design opportunity for a map-free localization system. Relative localization works for couriers' indoor localization because it can support multiple upper-layer applications. For example, Yang *et al.* [136] shows relative localization can be used to reduce indoor walking time in order dispatching.

To obtain the relative locations, we need to infer the distance between locations, which can be measured by travel time (i.e., an estimated time of arrival (ETA) problem). Compared to the existing work [87, 11] of estimating the end-to-end delivery time that mainly depends on the couriers' road travel time, our problem is focused on more granular time in the indoor environment, which depends on the "unobservable" indoor environment setting.

5.3.2 Encounters among Couriers

The value of the couriers' encounters is two-fold: (i) couriers' indoor mobility is patterned, so the encounters contain spatial-temporal information. For example, when two couriers encounter, we can use the time between a courier reporting "Departure" at a merchant and the encounter time as the travel time between the merchant and the encounter location because couriers usually take the shortest paths and there are no detours in between [136]. (ii) the encounters are dense; hence subsequent encounters have spatial-temporal connections. In an in-field experiment, we found that 79 couriers move around 37 merchants in a mall during rush hour (11 am), and more than 2,000 times of encounters are recorded, which introduces much more spatial-temporal information compared to reporting events only (around 200 reporting events). An illustrative example of the encounter event between two couriers is shown in Figure 5.1.

5.4 Design

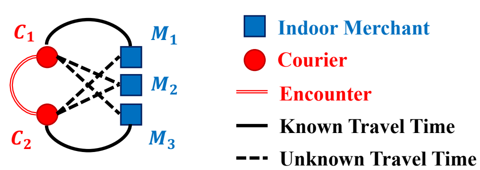


Figure 5.2: P^2-Loc Problem Setting

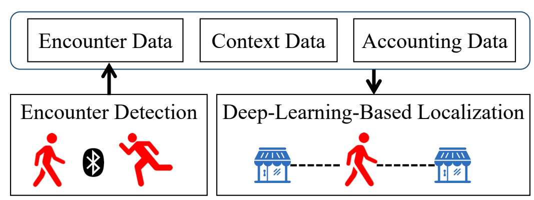


Figure 5.3: P^2-Loc System Overview

5.4.1 Problem Definition

The setting of the problem is shown in Figure 5.2. In a time-varying graph, the input includes the real-time encounter events between couriers (the double red line between courier C_1 and C_2), and the travel time between couriers and merchants (the solid black line between C_1-M_1 and C_2-M_3). The output is the couriers' real-time relative locations indicated by the travel time between the couriers and merchants (dashed black line between courier C_1, C_2, and merchants M_1, M_2, M_3).

Table 5.1: Encounter Data

Field	Value
Courier 1/2 ID	C001/C002
Enc. Start Time	7/1/20 12:11:00
Enc. End Time	7/1/20 12:11:20
Min./Max. RSSI	-90dB /-70dB
Var./Avg. RSSI	5.2 / -85dB

Table 5.2: Accounting Data

Field	Value
Order/Mer. ID	O001/M001
Accepting	7/1/20 12:00:00
Arrival	7/1/20 12:10:00
Departure	7/1/20 12:10:10
Delivery	7/1/20 12:25:00

5.4.2 Overview

Figure 5.3 shows the $\texttt{P}^2\texttt{-Loc}$ design with two modules.

Encounter Detection (Section 5.4.3). In this module, we detect the couriers' encounter events by (1) developing a BLE advertising and scanning module on couriers' smartphones; (2) mining the BLE data to extract encounter events. This mechanism is simple but robust given the real-world constraints, including privacy and security concerns (for both the platform and couriers), hardware compatibility (for both iOS and Android), robustness (fault-tolerant), and non-intrusive working manner (energy and data efficiency).

Deep-Learning-Based Localization (Section 5.4.4). We build a heterogeneous graph and conduct deep learning on the graph using courier-merchant embedding and merchant embedding to transform the heterogeneous nodes and edges into a unified space. The idea of link prediction in the recommendation system is used to predict the unknown travel time between couriers and merchants based on some known travel time and encounter information. Note that the model is dynamic based on periodically training with recent data.

Data. As indicated in Figure 5.3, three data-sets are collected to build the graph, encounter data, accounting data, and context data. In **encounter data** (Table 5.1), we calculate the statistics of the encounter events extracted from encounter detection. The **accounting data** (Table 5.2) logs the time and locations of four primary states of each order, i.e., accepting an order, arrival at the merchant, departure from the merchant (with the order), and final delivery to the customer. The state data are from couriers' manual reporting on their APPs. The accounting data are significant to the platform because they (1) are used for the platform's new order scheduling; and (2) are

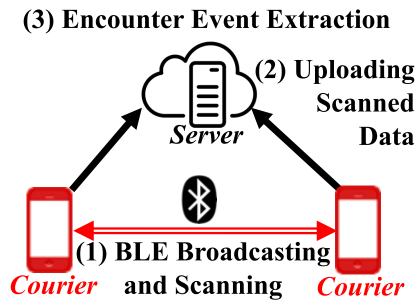


Figure 5.4: Encounter Detection

shown to customers in real-time to improve customers' experiences. The **context data** record some environmental information such as weather, date, and time.

5.4.3 Encounter Detection

Encounter detection, or proximity detection, has been studied using Wi-Fi [88, 180] and acoustic signals [181, 46]. These solutions, however, are not applicable in our setting. Wi-Fi is not applicable due to the scanning unavailability to non-iOS APPs in off-the-shelf iOS devices [182]. Acoustic is not applicable because of couriers' frequent use of microphones (to contact merchants and customers).

BLE Advertising and Scanning. We use the iBeacon protocol [183, 184] for BLE advertising and scanning, which is a connection-less protocol that does not need a pairing process. There are three parameters in the advertising `ID tuples`, a 16-byte `UUID`, a 2-byte `Major`, and a 2-byte `Minor`. As shown in Figure 5.4, the mechanism is as follows: (0) ask for couriers' consents that we can use their smartphones for encounter detection; (1) the consented couriers' smartphones conduct continuous BLE advertising and scanning at the same time in their working hours; (2) the couriers' smartphones upload the received `ID tuples` to a server in real-time by Internet connection (e.g., 4G); (3) the server extracts the encounter events from uploaded data.

The technical details in implementing the system are discussed in Section 5.5. A straightforward mechanism is used for advertising and scanning because (1) no additional configuration is needed after the couriers' initial consent, i.e., $\texttt{P}^2\texttt{-Loc}$ is transparent and non-intrusive to couriers; (2) APIs provided by Android [185] and iOS [186] are used to guarantee the compatibility of $\texttt{P}^2\texttt{-Loc}$, which leaves little design space for setting parameters such as transmission power and advertising cycle.

5.4.4 Deep-Learning-Based Localization

Motivation and Challenges of Using GNN. GNNs have made great success in graph learning tasks such as node presentation and link prediction. The strength of GNNs is in the ability to learn the structure information of graphs. It provides great potential to solve our problem because the links between couriers' locations and merchants' locations are unknown and different links are also strongly related. However, building $\mathtt{P}^2$-`Loc` based on GNNs has three challenges:
(i) the graph is heterogeneous (i.e., two types of nodes and edges, each with different features);
(ii) the graph is time-vary and temporally correlated because the edges are changing when couriers move around;
(iii) courier-courier encounter edges have multiple features such as encounter duration and RSSI statistics.

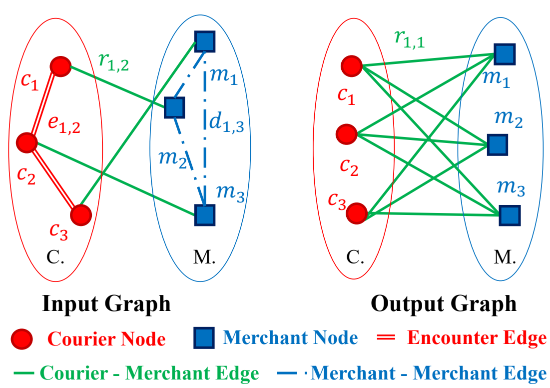


Figure 5.5: Input and Output

To address the challenges, we present a novel GNN framework with graph embedding to model heterogeneous nodes and edges, and a recurrent module to consider temporal correlations.

Notations. As shown in Figure 5.5, let $C = \{c_1, c_2, ..., c_{n_c}\}$ and $M = \{m_1, m_2, ..., m_{n_m}\}$ be the sets of couriers and merchants respectively, where n_c is the number of couriers, and n_m is the number of merchants. $E = \{e_{1,2}, ..., e_{n_c,n_c-1}\}$ is the set of courier-courier encounter event edges, and the data is collected from couriers' encounters. $R = \{r_{1,2}, ..., r_{n_c,n_m}\}$ is the set of courier-merchant travel time edges, and the data is collected from courier' reports and encounters when they travel from the merchants to the encounter locations, or from the encounter locations to the merchants; $D = \{d_{1,2}, ..., d_{n_m,n_m-1}\}$ is the set of merchant-merchant travel time edges, and the data is collected from couriers' reports when they travel among merchants.

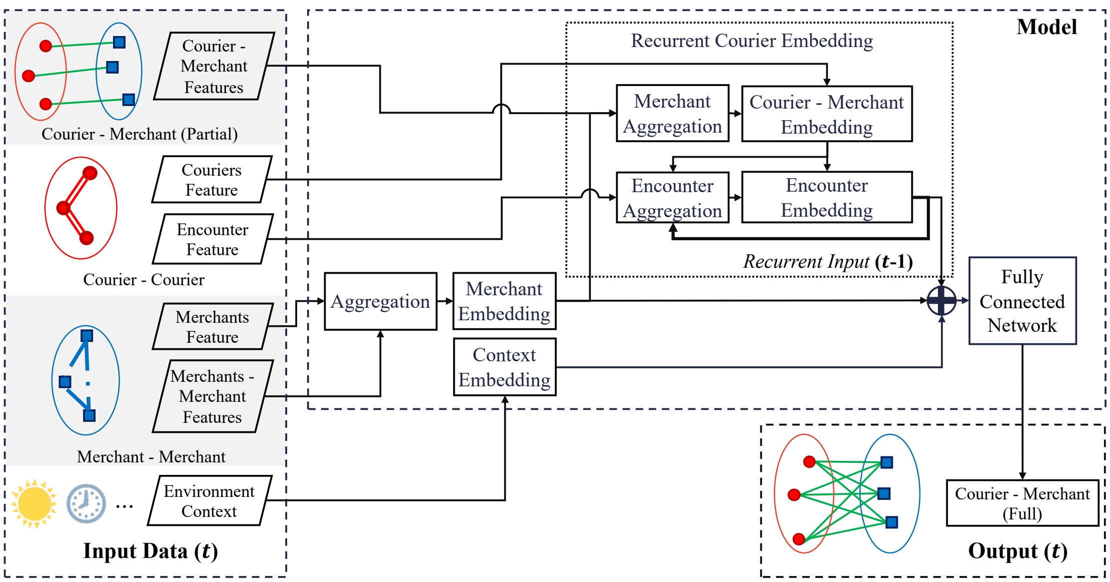


Figure 5.6: Deep-Learning-Based Localization Framework

Framework Overview. The architecture of the proposed model is shown in Figure 5.6. The model consists of four components: a recurrent courier graph embedding network, a merchant graph embedding network, a context embedding network, and a fully connected network. The input is a heterogeneous graph composed of three sub-graphs, i.e., an incomplete courier-merchant travel time graph, a courier-courier encounter graph, and a merchant-merchant travel time graph. We list all the node features and edge features in Table 5.3. The output graph is a complete courier-merchant travel time graph. Note that both the input and the output are time-based sequential data. In offline training, we use a time step t (e.g., 10s) to split the input data; in online predicting, we conduct the prediction at each time step.

Table 5.3: Features Used in Learning

Feature Category	Feature Name	Sample Value	Feature Category	Feature Name	Sample Value
Node Feature (Courier, C.)	Working Experience	2 years	Edge Feature (C.-C. Encounter)	Encounter Start Time	2020/07/01 12:00:00
	# of Picking Up Orders	2		Encounter End Time	2020/07/01 12:00:15
Node Feature (Merchant, M.)	Merchant Floor	1F		Encounter Duration	15 seconds
	Merchant Number	1F-23		Max. & Min. RSSI	-70 dB & -90 dB
	Merchant Type	Fast food		Avg. RSSI	-83 dB
	# of Preparing Orders	4		RSSI Variance	5.2 dB
Edge Feature (C.-M.)	Travel Time	12 seconds	Context Feature	Weather	Rainy
Edge Feature (M.-M.)	Travel Time	15 seconds		Rush Hour	Yes

Recurrent Courier Graph Embedding. The recurrent courier graph embedding aims to learn the latent factors and the temporal connection of couriers' relative locations. The challenge is how to combine the partial courier-merchant graph and courier-courier graph inherently. To address the challenge, we (1) conduct courier-merchant embedding with *Merchant Aggregation* for couriers to incorporate the travel time between the encounter location and merchant; (2) conduct encounter embedding with *Encounter Aggregation* for couriers to incorporate encounter information. Note that the output courier-courier embedding of the last time step is used as a recurrent input of the encounter aggregation to incorporate temporal connections.

In merchant aggregation, for a courier node c_i, we apply an aggregation function θ_R to its neighbors in the courier-merchants subgraph to generate the aggregated node feature $\mathbf{h}_i^R$. A merchants node is the neighbour of a courier node if it is the couriers' last-departed or next-arrived merchant, because they are on the short path to the courier's encounter location, and the travel time is known. Formally, we denote it as the following function:

$$\mathbf{h}_i^R = \sigma\left(\mathbf{W} \cdot \theta_R(\{\mathbf{x}_{i,j}, \forall r_{i,j} \in R\}) + \mathbf{b}\right) \tag{5.1}$$

where $r_{i,j}$ is an edge from courier c_i to a neighbour merchant m_j, $\mathbf{x}_{i,j}$ is a representation vector to denote the courier-merchant edge $r_{i,j}$. $\mathbf{x}_{i,j}$ is formulated as the concatenation of courier-merchant edge features (i.e., travel time) and merchant node embedding (after aggregation with its neighbors' information in the merchant-merchant graph); θ_R is the merchant aggregation function, and σ denotes the non-linear activation function (i.e., a rectified linear unit). $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias. One popular aggregation function for θ_R is the mean operator where we take the element-wise mean of the vectors in $\{\mathbf{x}_{i,j}, \forall r_{i,j} \in R\}$. This mean-based aggregator assigns equal weight to the couriers encountered, and we adopt a one-layer weighted mean-based aggregator in our implementation of θ_R [187].

$$\mathbf{h}_i^R = \sigma \left(\mathbf{W} \cdot \left\{ \sum_{r_{i,j} \in R} \alpha_{i,j} \mathbf{x}_{i,j} \right\} + \mathbf{b} \right) \tag{5.2}$$

where $\alpha_{i,j}$ is the weight of the encounter $e_{i,j}$, which can be an equal weight $\frac{1}{|E|}$ or an attention-based weight.

After the merchant aggregation, the courier-merchant embedding pipeline is used for concatenating merchant aggregation vector ($\mathbf{h}_i^R$) and courier features (C. in Table 5.3)) and feeding them into encounter aggregation and encounter embedding.

In the encounter aggregation, for a courier node c_i, we apply an aggregation function θ_E to its neighbors in the courier-courier subgraph to generate the aggregated node feature $\mathbf{h}_i^E$. Formally,

$$\mathbf{h}_i^E = \sigma \left(\mathbf{W} \cdot \theta_E(\{\mathbf{y}_{i,j}, \forall e_{i,j} \in E\}) + \mathbf{b} \right) \tag{5.3}$$

where $e_{i,j}$ is an edge (i.e., an encounter event) between courier c_i and c_j, $\mathbf{y}_{i,j}$ is a representation vector to denote the courier-courier edge $e_{i,j}$. $\mathbf{y}_{i,j}$ is formulated as the concatenation of the merchant aggregation vector ($\mathbf{h}_i^R$), courier-courier encounter information vector (C.-C. Encounter in Table 5.3), couriers attribute features vector (C. in Table 5.3), and the encounter embedding results from the last timestamp together as the input of the courier's embedded vector in the encounter aggregation. After the encounter aggregation, we concatenate the output of courier-merchant embedding and the output of encounter aggregation ($\mathbf{h}_i^E$) as the final output of the recurrent counter embedding.

Merchant Graph Embedding. The merchant graph embedding is used to learn the latent relationshop between the merchants (e.g., topology and relative locations) in a transformed spatial-temporal space. There are two steps: aggregation and merchant embedding.

In the aggregation, we generate the aggregated node feature for each merchant node by applying the aggregation to its neighbors in the merchant-merchant graph. Two merchants nodes are neighbors if a courier has traveled from one merchant to the other. The input includes two parts: merchants' feature set and merchant-merchant feature set. Merchants' feature set contains merchant-related features, such as the merchant floor, the merchant number, the merchant type, and the number of preparing orders, as shown in Table 5.3. Specifically, we feed sparse features of merchants to an embedding layer (the same as entity embedding, a simple lookup table). Then we concatenate the embedded sparse feature and dense features of merchants together as merchants' vector. The merchant-merchant feature set contains the travel time between merchants collected from couriers' historical travel time between merchants.

In the merchant embedding, we concatenate the output of the aggregation and the merchant feature (M. in Table 5.3) as the output of the recurrent counter embedding.

Context Embedding. Context embedding takes context information such as weather and time information as the input. Specifically, as the preprocessing of merchants features in merchant embedding, the context's representation is learned based on the entity embedding [188]. Then we fuse the entity embedding results of categorical features and the numerical features using concatenation operation.

Link Prediction. In the prediction part, the input is an arbitrary courier-merchant pair instance, and the output is the travel time between the courier-merchant pair. For a courier-merchant pair instance, we feed the concatenation of the courier embedding vector, the merchant embedding vector, and the context embedding vector to the fully-connect network. Then we get the regression result that stands for the travel time between the courier and merchant. We use ReLU as the activation function in the whole network architecture.

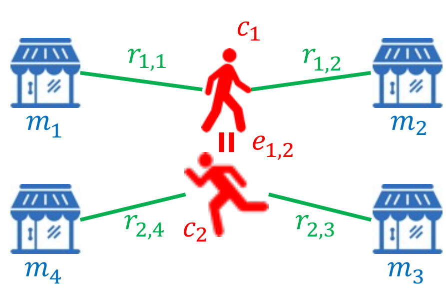


Figure 5.7: Training Illustration

Model Training. We adopt a common objective function

$$\text{Loss} = \frac{1}{|O|} \sum_{r_{i,j} \in O} (r'_{i,j} - r_{i,j})^2 \tag{5.4}$$

where O is a set of edges between couriers and merchants, $r'_{i,j}$ and $r_{i,j}$ are the predicted and labeled travel time between the couriers' encounter locations and merchants. A part (usually 20%) of the collected courier-merchant travel time data is used as labels, which is a common practice in the link prediction. Note that the labeled travel time is collected from couriers' manual reports when they arrive or leave the merchants. Therefore, no additional infrastructures are needed for collecting the label. We use a small example with two couriers and four merchants to show the process in Figure 5.7. When two couriers encounter on their ways among merchants, there are four courier-merchants edges and one courier-courier edge, and the corresponding data can be collected. In the training process, $r_{1,2}$ and $r_{2,4}$ can be used as labels since they are the travel time we want to predict (i.e., $r_{i,j}$ in (5.4)). Although couriers' reports only provide spares anchor information (i.e., two reports per order), we show that one-month data are enough to train the model with impressive performance. Admittedly, couriers' reports may have inaccurate data due to couriers' early or late reports, we use the Bluetooth beacons deployed in the mall to verify that couriers' reports work well as travel time labels.

Dynamic Environment. `P`^{2}`-Loc` is supposed to work in a dynamic environment where merchants' shops may open and close and couriers may come and leave. For the merchants' dynamics, since the model relies on the merchant-merchant graph embedding to learn the topology and relative locations of merchants in a mall, the merchants' information needs to be known in advance. Therefore, data need to be collected to train the model for new malls. The evaluation shows that one-month data are enough to train a satisfactory model that outperforms the baselines, which is marginal compared to the lifetime of a mall. The model also needs to be re-trained periodically to incorporate the new merchants and the floor plan changes (e.g., new elevators). For the couriers' dynamics, the courier's information does NOT need to be known in advance because the inherent information learned in the courier-courier graph and the courier-merchant graph is the relative distance between the merchants' locations and the encountered locations. When new couriers come, the prediction can be directly conducted as long as there are encounter events between the new courier and other couriers (i.e., the new courier is linked to the graphs).

Travel Time Asymmetry. In our design, we use travel time as a metric to measure the distance between a courier and a merchant, but in some extreme cases, the travel time between two locations is asymmetrical. For example, there might be a queue waiting for the upward elevator when the mall opens around 9 am, while the downward elevator is empty, which will lead to asymmetrical travel time between merchants on the ground floor and higher floors. In our graph embedding, we consider the asymmetry implicitly by assigning a direction for each edge.

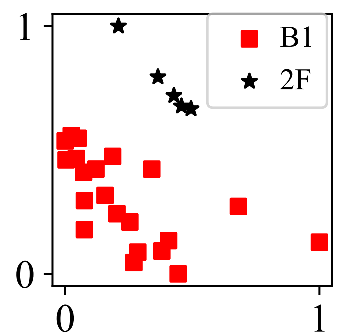


Figure 5.8: Embedding Visualization

Model Interpretation. To better understand the model, we visualize the merchant embedding. Figure 5.8 shows the projection of the embedding in a 2-D plane. The embedding learns both the floor information and distance information. The merchants on the same floor are nearby and the distance in the embedding is consistent with the travel time in between.

5.5 Implementation

The $\texttt{P}^2\texttt{-Loc}$ system was designed and implemented in the real world as a core component in the Ele.me platform, a large-scale on-demand delivery company in China. In the section, we introduce some practical issues rarely discussed or studied in a controlled environment.

5.5.1 Pilot Study and Real-World Deployment

In-Lab Pilot Study. We first conduct a feasibility study in the lab environment. We use five Android phones in the test and emulate the encounter events at five distances, i.e., 5m, 15m, 20m, 25m, 50m. We found that when the APP is active (either in the foreground or background), the advertising signal is stable within 15m with 90% encounter events captured, but degrades dramatically beyond 25m.

Large-Scale Real-World Deployment. After the in-lab study, we embedded the `P`^{2}`-Loc` into the couriers' APP in the [anonymous] platform. We developed two individual software development kits (SDK) for BLE advertising and scanning in the APP. We set some primary configurations of the `P`^{2}`-Loc` SDK as parameters for further developing, e.g., scanning duration and intervals, and data upload cycles. Note that the SDK and the back-end server developing were also the primary works in the implementation, but we omit them in this paper because they are standard.

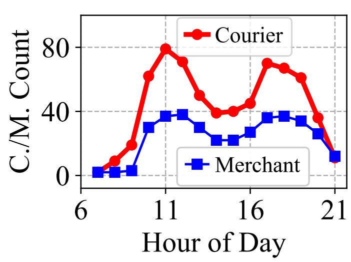


Figure 5.9: C./M. Node

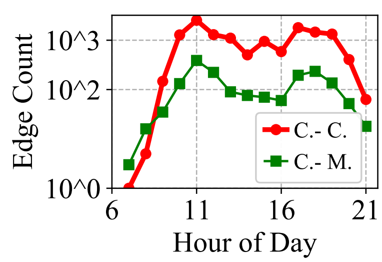


Figure 5.10: C.-C./C.-M. Edge

Real-World Encounter Facts. The `P`^{2}`-Loc` was embedded in the couriers' APP in June 2020, serving 4,008 couriers and 10,520 merchants. We use the data collected around a normal mall area to illustrate the statistics of the implementation. The average number of active couriers and merchants in different hours is shown in Figure 5.9, where we found that the number of couriers is twice of the number of merchants during the day. The average numbers of courier-courier encounters and courier-merchant interactions (i.e., picking up orders) are shown in Figure 5.10, where we found that the number of courier-courier encounters is ten times of the number of courier-merchant interactions during all the day. 87% of encounter events last less than 10 seconds (Figure 5.11), and almost all the encounter events (99%) last less than 55 seconds.

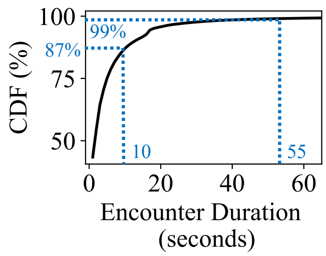


Figure 5.11: Duration CDF

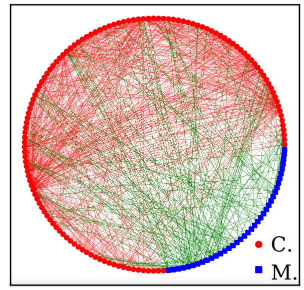


Figure 5.12: Input Graph at 11 am

The courier-courier encounter and courier-merchant graph in the rush hour (11 am) in a single day are shown in Figure 5.12. Red circles stand for couriers, and blue squares stand for merchants. Red lines are encounter events between couriers, and green lines are known travel time between couriers and merchants. In the rush hour, 79 couriers move around 37 merchants, making 389 courier-merchants interactions and 2,534 courier-courier encounter events. During the non-rush hour (4 pm), 45 couriers move around 27 merchants, making 60 courier-merchants interactions and 588 courier-courier encounter events. Intuitively, the encounter density impacts the localization performance, and we will show its impact in Sec. 5.6 (Figure 5.22).

5.5.2 Reliability of Encounter Detection

In the implementation, we found that not all encounter events can be detected by our encounter detection module, hence we conducted some studies to find out the reasons. Specifically, we define

$$\text{Encounter Detection Reliability} = \frac{\text{\# of Detected Encounters}}{\text{\# of Total Encounters}} \tag{5.5}$$

where the number of total encounters is estimated based on the BLE beacons we deployed at merchants (Figure 5.18) and in-field observation (we spent two days in the mall and record the encounters of couriers).

Impact of Encounter Duration. The encounter duration, calculated as the first and last BLE advertising timestamp, has a significant impact on the encounter detection reliability. As shown in Figure 5.13, it can be observed that reliability is less than 80% when the encounter duration is less than 25s and is greater than 90% when the encounter duration is longer than 50s.

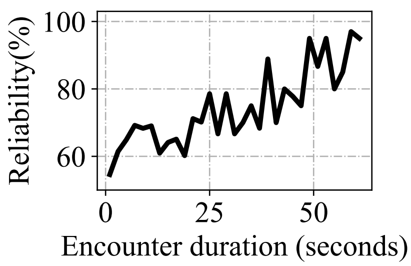


Figure 5.13: Duration Impact

Table 5.4: Device Impact

Broadcast Device (shares) \ Scan Device (shares)	HUAWEI (35%)	Xiaomi (17%)	OPPO (12%)	Vivo (32%)	Others (4%)
HUAWEI (44%)	33%	38%	43%	34%	53%
Xiaomi (23%)	64%	77%	79%	78%	76%
OPPO (17%)	78%	77%	86%	82%	76%
Vivo (16%)	70%	75%	64%	75%	64%

Impact of Device Hardware. Because there are 52 brands and 672 phone models used by 300K daily active couriers, the system must be compatible with most (if not all) devices. We illustrate the reliability between four major brands as advertising and scanning device pairs in Table 5.4. Different devices show significant differences as advertising devices. For example, HUAWEI shows inefficiencies compared with other brands when advertising, possibly due to hardware or software differences. We show that $\texttt{P}^2$`-Loc` works robustly given undetected encounter events (Figure 5.21).

5.5.3 Privacy in BLE Advertising

Potential Privacy Weakness. In the iBeacon protocol [183] we used, an `ID tuple` is fixed for each device, and the advertising is in *cleartext*. It leads to courier privacy and platform security issues under potential attacks [189, 190, 191]: (1) an adversary can replicate some courier `ID tuples` and advertise them in some other locations, which can lead to wrong encounter detection and problematic order assignment; (2) an adversary can deploy some devices to eavesdrop on the couriers' `ID tuples` and record their advertising locations as "side information" through war-driving [192]. The side information is used to attack an anonymous open data-set (e.g., anonymous online reviews or leaked data from the platform) to re-identify certain couriers [193].

Privacy Protection Mechanism. To address the potential privacy problems, we augment our advertising with the SM3 algorithm, i.e., a public Time-based One-Time Password (TOTP) [194] algorithm to encrypt the `ID tuples`, similar to Google Authenticator. Specifically, the server assigns a seed ID to a courier's phone when he logs in to the platform using the smartphone for the first time. For every duration of K, the server conducts the following three steps: (1) calculating an encrypted `ID tuples` for each smartphone based on its seed ID and current time; (2) updating the mapping of the courier's identity and its newly encrypted `ID tuples`; (3) sending the encrypted `ID tuples` to the smartphone for advertising in the following duration K.

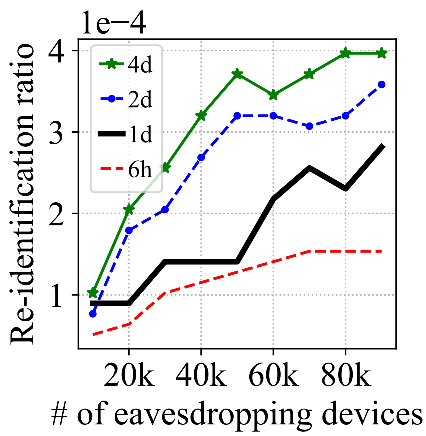


Figure 5.14: Privacy Risk

Analysis Results. We evaluate the performance of our location privacy protection with offline analysis on real-world Bluetooth and trajectory data. In the attacking model, a group of adversary devices is randomly deployed at known locations to eavesdrop on couriers' advertising messages to collect side information. The side information is used to attack a supposedly-leaked anonymous data-set with all couriers' traces. In the experiments, we assume adversary devices were deployed in a group of merchants with known locations, then we find the couriers around the merchants as the eavesdropped couriers based on couriers' trajectory. We use 78.1K couriers' trajectory data in one day in Shanghai with anonymous ID as the supposedly-leaked data, and compare it with the couriers' location we eavesdropped around the merchants in a brute-force way. Based on the [193], four spatial-temporal points are enough to identify most individuals. In Figure 5.14, we show that the re-identification ratio increases as the adversary utilizes more eavesdropping devices. The re-identification ratio is defined as how many couriers we can identify successfully among all couriers. When we set the ID update cycle as one day (default setting), we found that the possibility of a courier getting re-identified is less than 0.03% even 80k eavesdropping devices are deployed. The risk ratio is still below 0.04% when we use four days as ID update cycles.

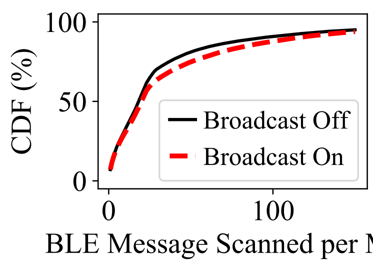


Figure 5.15: Impact on BLE Scanning

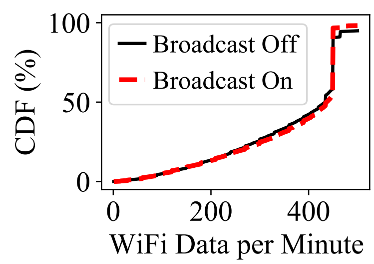


Figure 5.16: Impact on WiFi Scanning

5.5.4 Other Real-World Issues

BLE Advertising - Scanning Interference and BLE - Wi-Fi Interference. One potential problem in implementing $\texttt{P}^2\texttt{-Loc}$ is that there might be conflicts between BLE advertising and scanning. We test the interference by comparing the number of BLE advertising messages scanned when the BLE advertising is on and off. The data indicates that the advertising and scanning module does not conflict because the number of BLE messages scanned per minute does not change much when the advertising is on and off (Figure 5.15). We also test the interference between BLE and Wi-Fi by comparing the number of Wi-Fi data scanned when BLE advertising is on and off, and no interference is observed (Figure 5.16).

APP and Service Lifetime. We also find a key factor that impacts the encounter detection performance (and possibility all the APP-level mobile computing applications) is that the operating system may kill the BLE advertising service due to resource or energy reasons. In most cases, multiples SDKs are packaged in an APP, and each is operating different tasks. Android [121] suggests that consistent tasks without user intervention should be implemented using "service", so our SDK registers a service whenever the APP initiates. However, the registered service might be killed by the OS without notification; hence the SDK is unaware when the BLE advertising or scanning fails. One solution is to re-start the service periodically to avoid unknown failures, which is simple but cannot adapt to different cases. We implement an adaptive strategy that dynamically re-start the service according to the courier's status (indoor/outdoor, order status, etc.). We omit the details here due to space limitations.

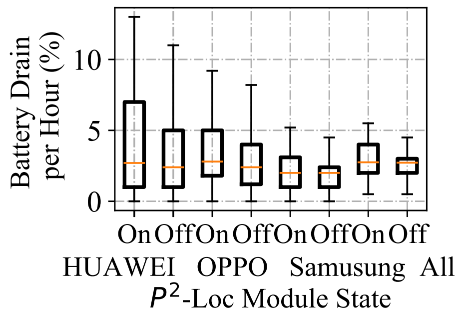


Figure 5.17: Energy Consumption

Energy Consumption. In a small-scale test, we asked 62 couriers' permission for their smartphone battery information to analyze the impact of $\texttt{P}^2\texttt{-Loc}$ on smartphone energy consumption. We collected the battery level data from the Android API [195] and removed the corresponding data if the phone is charging. Then we grouped the data by $\texttt{P}^2\texttt{-Loc}$ states (on/off) and brands and calculated the battery drain per hour. The median battery drain per hour is 2.72% when $\texttt{P}^2\texttt{-Loc}$ is off and 2.75% when $\texttt{P}^2\texttt{-Loc}$ is on. The absolute additional energy consumption is 0.03% per hour, and the relative additional energy consumption is 1%. The boxplots of battery drain per hour of all smartphones and some brands are shown in Figure 5.17. The results show that different smartphones show different energy consumption, but the additional energy consumption of $\texttt{P}^2\texttt{-Loc}$ is marginal.

5.6 Evaluation

5.6.1 Methodology

Settings. We conducted a thorough evaluation of $\texttt{P}^2\texttt{-Loc}$ in a mall area in Shanghai in Figure 5.18 for one month (June 2020). There are 294 active couriers, 51 active merchants, and 2427 delivery orders each day. We deploy some BLE beacon devices in the merchants to get ground truth.

Metrics. In $\texttt{P}^2\texttt{-Loc}$, we achieve indoor relative localization from the couriers to the merchants. To evaluate the relative location inferred by $\texttt{P}^2\texttt{-Loc}$, we compare the estimated travel time between the encounter location and a merchant with the ground

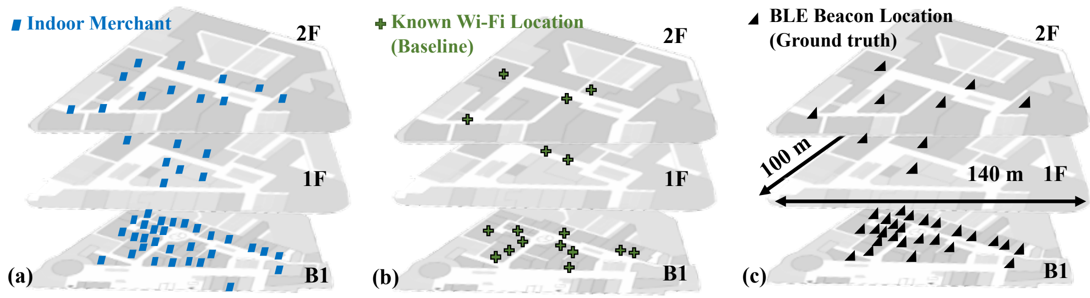


Figure 5.18: The three-floor mall in the evaluation. The goal is to infer couriers' relative locations to all the merchants (Figure(a)). Wi-Fi APs with known locations are used in the Wi-Fi-based baseline (Figure(b)). Ground truth is collected via BLE beacons deployed at some merchants (Figure(c)).

truth that the courier visited after. We define an absolute time error (ATE) as follows

$$\text{ATE} = |t_i - \hat{t}_i| \tag{5.6}$$

where $|\hat{t}_i|$ and t_i are predicted travel time and the ground truth respectively.

Ground Truth. Although the training labels can be collected from couriers' reports, a question remains that whether the collected labels work well in a real-world application given potential inaccuracies due to couriers' early or late reports. Therefore, we use the BLE beacons deployed at the merchants to get the ground truth for the couriers' travel time among merchants. We obtained 34 merchants' consents to deploy BLE beacon devices, and each beacon device is bound to one merchant, as shown in Figure 5.18. We follow the idea in [196] to find the accurate time of couriers' arrival and departure at each merchant; hence we know the ground truth travel time from the encounter location to the merchants. The training process is infrastructure-free since only couriers' report is used, and we use the ground truth data in the testing phase not only to evaluate the model but also the effectiveness of couriers' report as labels. Since we have verified the effectiveness of couriers' reports as labels in the work, no hardware or infrastructure when we apply `P`^{2}`-Loc` in real-world applications.

Baselines. We choose the following baselines and group them into two categories: *anchors baselines* and *model baselines*. We use anchors baselines including Wi-Fi, GPS, TransLoc, and $\texttt{P}^2\texttt{-Loc-}$ to show the effectiveness of utilizing encounters as additional information. We use model baselines including MDS and Statistical methods to show the effectiveness of GNN. We do not include the methods that required sophisticated devices or extra fingerprinting effort, considering their practical constraints in large-scale deployment.

• *Wi-Fi-based Methods (WiFi).* To show $\texttt{P}^2\texttt{-Loc}$'s superiority compared to state-of-art Wi-Fi-based methods, we embedded a Wi-Fi scanning module in some couriers' Android phones APP under their consent. The module periodically scans the Wi-Fi signals around and returns the Wi-Fi list with a timestamp. The scanning cycle is set as one minute, but the scanning would be throttled when the APP runs in the background [197]. We follow the idea in [198], where we use the order of Wi-Fi RSSI values to find the courier's location and calculate the $\hat{t}_i$ of Wi-Fi based on the couriers' historical travel time. We conducted a wardriving process and collected the location of 18 Wi-Fi APs (Figure 5.18).

• *GPS-based Methods (GPS).* To show $\texttt{P}^2\texttt{-Loc}$'s superiority compared to state-of-practice methods. We utilize received GPS signals to localize couriers. The $\hat{t}_i$ of GPS is calculated using the distance between the courier and the merchant, and the average speed variable depends on the area and time.

• *Report-based Methods (TransLoc).* To show the advantages of combining encounter data and accounting data with using accounting data only, we implement TransLoc [136] where they build a symbolic graph from couriers' report data to predict the courier's arrival time at the indoor shops.

• *No-Encounter ($\texttt{P}^2\texttt{-Loc-}$).* To show the effect of encounter information, we implement a deep learning model without encounter data. That is, we only use the courier-merchant distance and the merchant-merchant distance to build the graph in Figure 5.5 and Figure 5.6.

• *Encounter-based Localization Methods (Enc-MDS, Enc-Stat.)* To show the effectiveness of using GNN in encounter-based localization, we consider two encounter-based works using multidimensional scaling (MDS) [169] and statistical method [174]. The statistical method (Enc-Stat.) is a hybrid solution where Wi-Fi is used for localization and Bluetooth is used for encounter detection.

Hardware and Parameter Settings. The detailed parameter settings for GNN

Table 5.5: Parameter Settings.

Hyper-parameter	Setting	Hyper-parameter	Setting
GNN hidden dims.	64	Batch size	512
Embedding dims.	3	Learning rate	0.001
FC network hidden dims.	64	Optimizer	‘RMSprop’
Predict time step	10 seconds	Activation	‘Relu’

are provided in Table 5.5. The parameter settings are based on the tradeoff of couriers' indoor mobility, order batching and scheduling cycle, and computation cost. The following hardware and software configurations are used in the evaluation: CentOS, NVIDIA GeForce RTX 2080 Ti, and 78G memory.

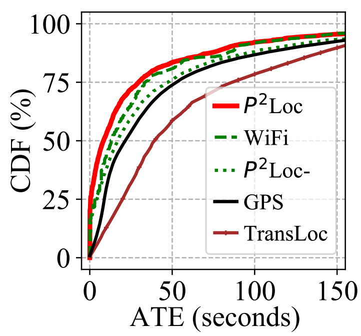


Figure 5.19: Performance of Different Types of Anchor Info.

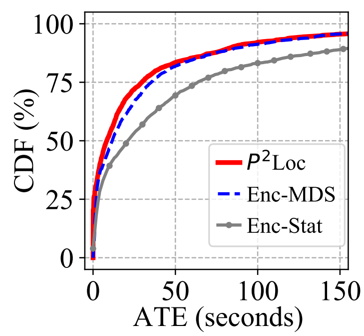


Figure 5.20: Performance of Different Models

5.6.2 Results and Analysis

Performance Compared with Different Types of Anchor Information. Figure 5.19 shows the performance of `P`^{2}`-Loc` compared with anchor baselines, where `P`^{2}`-Loc` performs better than all the baselines consistently. Quantitatively, the mean absolute error (MAE) in seconds are 29.39s for `P`^{2}`-Loc`, 32.41s for Wi-Fi, 39.65s for `P`^{2}`-Loc-`, 36.13s for GPS, and 60.40s for TransLoc. It shows that `P`^{2}`-Loc` improves methods based on Wi-Fi, GPS and reporting methods by 9%, 19%, and 51%, respectively, bearing out the advantage of utilizing encounter information compared to other anchor-only solutions. The improvement compared to `P`^{2}`-Loc-` (26%) also verifies the value of encounters in a deep-learning-based method.

Performance Compared with Different Models. Figure 5.20 shows the performance of $\texttt{P}^2\texttt{-Loc}$ compared with model baselines, where the MAE are 32.04s for Enc-MDS, and 42.61s for Enc-Stat. The improvement introduced by $\texttt{P}^2\texttt{-Loc}$ is 8% for Enc-MDS, and 31% for Enc-Stat, bearing out the advantage of using GNN.

Performance versus Cost Tradeoff Analysis. We analyze performance (i.e., accuracy) versus cost based on different system design choices: (i) exploring both new hardware and new software (i.e., aBeacon [1]); (ii) new software only (i.e., our $\texttt{P}^2\texttt{-Loc}$); (iii) neither new hardware nor new software (i.e., TransLoc [136]). Here the hardware and software are additionally deployed or developed instead of existing components such as smartphones or the delivery APP. We argue $\texttt{P}^2\texttt{-Loc}$ can achieve better accuracy and cost tradeoff compared to other design choices. Compared to aBeacon with 80% accuracy on average [1], $\texttt{P}^2\texttt{-Loc}$ can achieve the same accuracy if we consider 46 seconds as the threshold (shown as in Figure 5.19), but much less costly than aBeacon (i.e., \$10 for hardware only for each shop excluding installation). Even we set the threshold to be 30 seconds (i.e., the pooling time for order dispatching), the accuracy is 74% with a limited decrease compared to aBeacon. Compared to TransLoc with the lowest cost, $\texttt{P}^2\texttt{-Loc}$ doubles the accuracy on average (shown as in Figure 5.19) but only has a nearly neglectable cost of software development.

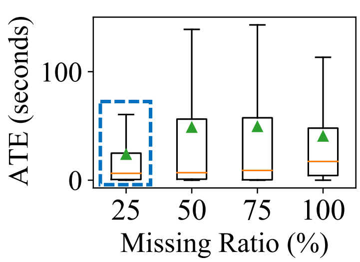


Figure 5.21: Robustness

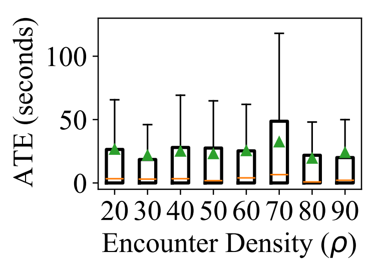


Figure 5.22: Encounter Density

Robustness. Given the reliability issues in encounter detection (Section 5.5.2), we realize that not all encounter events can be captured. We evaluate the robustness of $\texttt{P}^2\texttt{-Loc}$ by setting a part of encounters as "undetected" (i.e., corresponding encounter data not used), and show the performance at different missing ratios in Figure 5.21. Note that the "undetected" encounter data are selected based on a uniform distribution to mimic the real-world setting. Compared with real-world cases (around 25% missing in dashed box), the median ATE (brown bar in the box) does not change much when more encounters are missing, but the mean ATE (green triangle) increases when more than 50% of encounter events are NOT detected. This suggests the importance of a reliable encounter detection mechanism. We envision that better localization performance can be achieved with higher encounter detection recall brought by updated smartphone hardware and encounter detection modules.

Impact of Encounter Density. We also evaluate the impact of the density of couriers, merchants, and their interactions by comparing the performance at different hours. The density ρ is defined as the number of average encounter events per courier per hour. Note that unlike the missing encounters in robustness evaluation, when the density decreases, the number of encounter events, the number of couriers, merchants, and their interactions also decreases. As shown in Figure 5.22, MAE varies between 19.5s ($\rho = 80$) and 32.3s ($\rho = 30$) when density varies between 20 and 90. A less apparent observation is that MAE decreases when the density increases, but the MAE when $\rho = 30$ and 70 deviates from this trend due to the limited evaluation scale (i.e., one mall). Note that the overall density is around 40, where the MAE is around 27s.

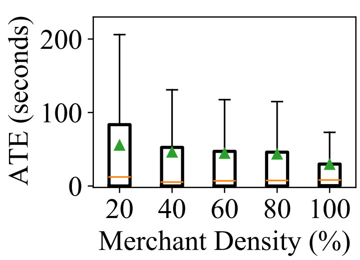


Figure 5.23: Merchant Density

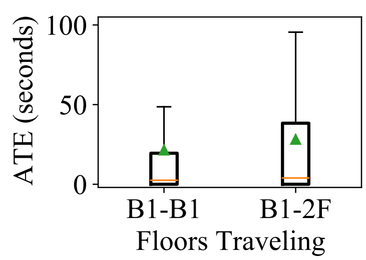


Figure 5.24: Impact of Floors

Impact of Merchant Density. The density of merchants in the mall also impacts the performance, because the couriers' reports at the merchants are used as anchor information. Therefore, the merchant density can also be viewed as the anchor density in the evaluation. We do the evaluation by manually selecting certain ratio of the merchants in the training and testing. As shown in Figure 5.23, the x-axis is the ratio of merchants we used in the evaluation, where 100% means all the merchants are used. It can be observed that the performance degrades when there are fewer merchants in a mall. The degradation is limited when the density varies between 40% and 80%, but will drop significantly when the density decreased to 20%. Note that the mall in the evaluation has 51 merchants located in three floors where each floor cover an area of 8000 square meters.

Impact of Encounter Merchant Floor. Unlike other indoor localization problems, one challenge in our setting is that couriers travel between indoor merchants on different floors. We evaluate the impact of cross-floor travel by extracting the same-floor data (B1-B1) and the different-floor data (B1-2F). As shown in Figure 5.24, the MAE for the same floor and different floors are 21.26 seconds and 27.91 seconds, respectively. It indicates that cross-floor localization is more challenging than same-floor localization, but we can still provide better performance than Wi-Fi and GPS.

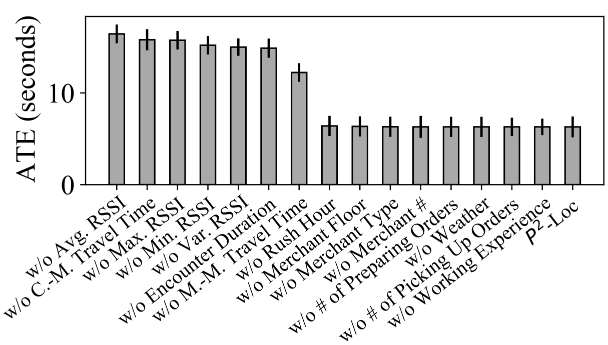


Figure 5.25: Feature Sensitivity Analysis

Feature Sensitivity Analysis. To evaluate the feature importance, we conduct a sensitivity analysis via the leave-one-out method (i.e., training the model without a feature). The complete model (i.e., $\texttt{P}^2\texttt{-Loc}$) with all features is used as the baseline. As shown in Figure 5.25, edge features such as RSSI statistics, encounter duration, C.-M. travel time, and M.-M. travel time contribute mostly. The contribution of context features and node features is limited due to the limited evaluation environment (one mall) and time (one month). We envision these features will make more contributions in a more extended time scope and extensive area.

Model Latency and Update, Scalability, Online Deployment, and Model Generalization. For the learning model, the offline training takes 4 hours and the online prediction takes 5 seconds. Although the travel time among shops does not vary much day by day, it varies slowly when couriers enter and leave a delivery team, shops open and close in a mall, floorplan changes, and dispatching policy changes. Therefore, the models are updated periodically (e.g., weekly) with new encounters and couriers' report data. Although our work is based on an offline experiment, the model can be directly deployed online given the cloud services (e.g., Tensorflow on Google Cloud [199]). The scalability of the model can be achieved by parallel training and prediction in different malls or districts. Note that because the graph model relies on the merchants' locations and couriers' mobility, it is inappropriate to use an existing model on new malls and couriers. However, we have shown that one-month data is enough to build a decent model, and one month is a short time compared with the alteration of merchants and couriers.

5.7 Discussion

Key Lesson Learned: Trade-off Between Performance and Model Complexity. The results (Figure 5.19, 5.20) suggest that the deep learning based approaches outperform naive GPS and simple mechanisms (i.e., MDS) moderately, leading to a 7-second and 3-second performance gain, respectively. This leads to a set of important lessons learned, especially for industrial applications, that (1) a simple approach (e.g., GPS) could be adopted first because it is easy and cheaper to deploy; (2) some advanced approaches (e.g., MDS) can improve the performance given additional information that can be obtained in a low-cost fashion as inputs (e.g., encounter data); (3) an approach based on complex deep learning may further improve the performance with little performance gain but requires massive training data with accurate labels (that might not be easy to collect at a large scale) and fine-tuned parameters (that may fail when the environment changes). Thus, how to achieve a balanced tradeoff is based on specific business requirements: when low-cost additional information (e.g., encounter data) is available, some simple data-driven approaches can be used; when labeled data can be obtained at a low cost and even a moderate improvement is also important (in our case the localization service helps the platform schedule 10 million orders each day), some deep learning approaches could be considered to push the performance to the limit.

Other Lessons Learned. (1) The courier's indoor encounter events offer a great opportunity to achieve relative infrastructure-free indoor localization (Figure 5.1). (2) Graph learning works well for the indoor localization problem because the indoor topology and spatial-temporal connections between couriers and merchants can be represented and learned effectively and efficiently (Figure 5.8). (3) We show that the system is non-intrusive, energy-efficient, and privacy-preserving in the large-scale real-world implementation (Figure 5.14, 5.15, 5.16, 5.17). (4) Although the reliability of the BLE-based encounter detection is impacted by multiple factors (Figure 5.13, 5.4), we show that $\texttt{P}^2\texttt{-Loc}$ is robust (Figure 5.21), and potential improvement is expected with hardware updating. (5) We show that $\texttt{P}^2\texttt{-Loc}$ outperforms anchor-based methods (i.e., GPS, Wi-Fi, and TransLoc), and encounter-based methods without deep learning (Figure 5.19, 5.20).

Further Explanation on "Anchors". "Anchor information" in this paper means location information from any source, and it is indispensable in a localization system. While for existing "anchor-free" works, they either provide "infrastructure-free" solutions [200] as in our paper or build a "target-relative" coordinate system where only relative locations are needed [201], which does not work in on-demand delivery since we need to know couriers' relative locations to the merchant shops, instead of to other couriers. In $\texttt{P}^2\texttt{-Loc}$, we used couriers' reports at the merchants as semantic anchor information. Compared to infrastructures, the major drawback of couriers' reports as semantic anchors is that the report is sparse because couriers only report twice (arrival/departure) during the whole delivery process. Therefore, we explore the additional spatial-temporal information in couriers' encounters for localization.

More Applications based on Encounters. Encounter-based indoor localization is our first application, and more applications are envisioned based on encounter detection in on-demand delivery. For example, in the current delivery scheme, an order is delivered by a single courier. This strategy is simple for scheduling and accounting but may have lower efficiency, especially when multiple couriers are waiting at the same merchant or traveling between similar routes. One potential improvement is that we check if we can swap their orders or put all the orders to one courier and free another courier when two couriers encounter.

Generalization to other Applications. Although $\texttt{P}^2\texttt{-Loc}$ is designed and implemented in on-demand delivery, we believe the system and the underlying ideas work in generic scenarios where relative locations are needed based on a few known spatial-temporal points and some encounters. For example, warehouse robots [202] have been envisioned for many years but are still not widely applied due to the high cost. One of the costly parts is the onboard sensors for localization. We believe $\texttt{P}^2\texttt{-Loc}$, with some modifications, is a potential solution to provide an accurate yet low-cost localization service. Another potential application is vehicle-to-vehicle communication, such as dedicated short-range communications (DSRC) [143]. We can infer all the vehicles' locations given their short-range communication and a few vehicles with known locations.

Limitations. (1) *No Absolute Indoor Localization.* In this work, we infer a courier's relative distances to the indoor merchants as the courier's relative location. We argue that this "relative location" is enough for the on-demand delivery because knowing the distance between the couriers and the merchants is enough for order scheduling and time estimation. For some other applications, absolute locations are needed, so we usually need the floorplan or the absolute location of the anchors (e.g., Wi-Fi AP, BLE Beacon, LED light) to acquire the absolute locations of the target users. However, the floorplan is not always available for all the malls in a city. (2) *Cross-Mall Application.* Since the merchant topology is learned in the merchant embedding, the model only works in the mall where the labeled data are collected. Data collection and model training are needed when we want to apply the model in a new mall. (3) *Inaccurate Courier Report.* It has been shown in [136] that couriers' reports are prone to errors. This paper adopts similar pre-processing as [136] on the report data to filter out errors. The results (Figure 5.19) show that solution solely relying on reporting data performs badly. However, integrating massive encounter data with reporting data can significantly improve localization results (Figure 5.19).

Chapter 6

Decision-Making for Human-System Synergy

6.1 Introduction

A key part of CPS that differentiate other systems that are purely based on sensing and prediction is that in CPS, decisions made based on sensing and prediction are applied in the physical world and influence the physical process. Particularly in a human-centered CPS application, the decisions usually impact humans. This introduces research problems in two directions: (1) how humans help systems have better performance (e.g., operate more efficiently); (2) how systems help humans do more things (e.g., automation). Compared to the second problem that has been the target for almost all the computer science problems, the first problem becomes more interesting in the big trend of human-AI collaboration [203].

In this chapter, we will use two examples to show (1) how we build an early arrival notification mechanism to help improve the system performance with the feedback from couriers based on the beacon system introduced in Chapter 4; (2) how we apply the results from travel time prediction to improve the order dispatching efficiency based on the $\texttt{P}^2\texttt{-Loc}$ introduced in Chapter 5.

6.2 Early Arrival Notification

6.2.1 Background

In gig delivery, some platforms (e.g., Eleme, Meituan) set a deadline for each order and promise refund if the order is overdue. To set a reasonable deadline for each order, the platform first collects historical data from multiple sources including the order log data (e.g., timestamp for order placement and dispatching), couriers' manual report on order progress (e.g., arrival at the merchant, delivery at the customer), couriers' trajectories from different sources (e.g., GPS, Wi-Fi, Bluetooth). With this data, the platform builds a prediction model to predict the delivery time for each order [87]. Then the delivery deadline will be calculated by adding some margin (e.g., 3 minutes) to the predicted delivery time. After that, the courier will try to deliver the order within the deadline. In this process, the courier also needs to report the arrival time and the delivery time of this order and the report will be used to predict the delivery time for future orders.

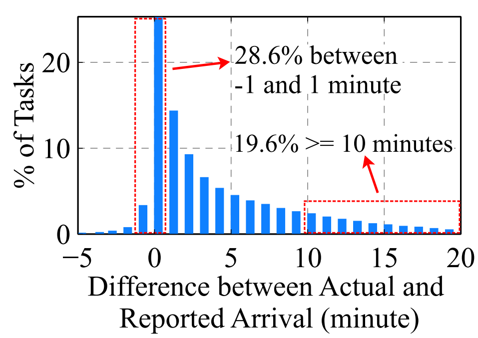


Figure 6.1: Early Arrival Reporting at the Merchants

However, with the beacon system, we found a problem – early arrival reporting at the merchants – in this process. That is, the courier usually reports the arrival before they actually arrive at the merchant. Figure 6.1 shows the distribution of the reporting time error, which is calculated as the difference between the actual and the reported arrival time. Only 28% of the orders have accurate arrival time reported within 1 minute of the actual arrival. In most cases, the couriers tend to report arrival early, and for 19% of the orders, the courier even reports arrival more than 10 minutes before they arrive. Based on some surveys from the couriers, we realize there are two main reasons for couriers to do so. The first is to avoid penalties, because once the courier report arrival at the merchant, he implies that he has been there waiting for the order, and if the order is overdue, the platform will penalize the merchant instead of the courier. The second is to attract more orders, because the courier wants to show that they are more efficient than others so that they will have more orders in the future. However, because the time they reported will also be used by the algorithm to decide the future deadline, there will be severe consequences of the early reporting. The first is the unreasonable short delivery deadline. Because the algorithm is misinformed, it will set the deadline to be shorter than that can be completed. In the past three years, the average delivery time in China has been reduced by 10 minutes. It also leads to increased traffic offenses of couriers due to speeding, and here are 2000 traffic offenses one month in one city in China [204]. Therefore, we need solutions to get out of this loop. There are two directions, The first is to collect accurate data without couriers' participation, as we have done in the physical and virtual beacon system, but its coverage and accuracy are still limited. The second is to collect accurate data by impacting couriers' behavior.

6.2.2 Notification on Early Reporting

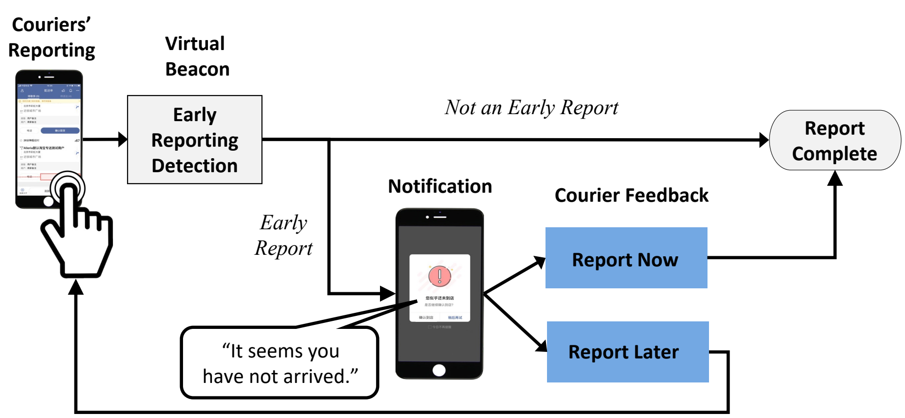


Figure 6.2: Early Arrival Notification Mechanism

We design an early reporting notification mechanism, which works as shown in Figure 6.2 When the couriers report arrival at the merchant, the virtual beacon system will decide whether it's an early reporting. If it is not an early report, the report is completed. Otherwise, if it is an early report, the platform will pop up a notification and tell the courier “It seems you have not arrived”. Then the courier can choose to insist on reporting by clicking “Report Now” or withdraw this report by clicking “Report Later”. If the courier chooses “Report Now”, we will trust the courier and complete the report. If the courier chooses “Report Later”, he will need to go through this process again when he wants to report arrival later.

6.2.3 Results and Analysis

Although we have this notification mechanism, there is still a problem: the platform may give incorrect notifications due to false negative detection because virtual beacon is not 100% reliable as discussed in Section 4.3.1. Specifically, when the platform sends a notification to the courier, the ground truth can be that the courier does not arrive, or the courier actually arrives. When the courier does not arrive, this is a correct notification; when the courier arrives, this is an incorrect Notification. For each kind of ground truth, the courier can choose to report now or report later, so there are four cases in total. In the four cases, we are interested in two cases. In the first case, if the courier arrives and he chooses to report now, we can use it as feedback to improve the system performance because it indicates there is a false negative detection in the virtual beacon system. In the second case, if the courier does not arrive and he chooses to report later, the courier's behavior is changed. For the other two cases, we are not sure of the reasons, and we will study them in future works. Note that the ground truth is based on the posterior analysis of couriers' trajectory with accurate GPS, which can be obtained in some cases but are not widely available.

In the first case, when we have an incorrect notification to a courier who has arrived, the couriers click report now to help us find some unexpected cases. Our results show that after two months, the ratio of “Report Now” is around 67%. Based on this feedback, we perform in-depth system analysis to improve the system performance. The incorrect Notification drops from 56% to 42%. It indicates the system performance is improved since we have fewer incorrect notifications.

In the second case, when the system makes the correct notification, we expect the courier to click report later, which means their behavior is changed. Our results show that after around two months, there are around 30% of the feedback is “Report Later”. That means around 30% of the reporting behavior is changed. We also measure this

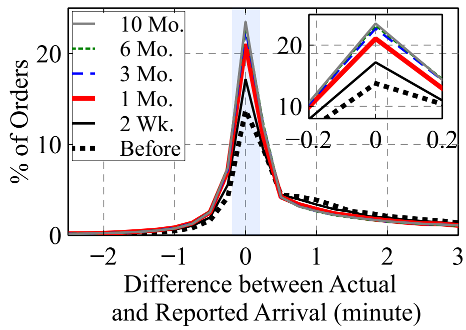


Figure 6.3: Reporting Behavior Change

change by the average reporting time error. To show the behavior change, we show the results of two weeks, 1, 3, 6, and 10 months after the notification was added. We found that the longer the notification was added, the higher percentage of orders with the difference closer to 0, indicating the couriers have been improving their reporting behavior. In particular, the percentage of reporting with errors smaller than 30 seconds increases from 36.1% to 49.5% after a three-month nationwide intervention, while increasing subtly to 50.3% after 10 months. It indicates the marginal effect of intervention decreases after a long time. Both figures show that the couriers' behavior is changed since the report time error is reduced. With more accurate data collected, the algorithm and the system can better serve merchants, customers, and couriers.

Based on the two cases here, we also have an interesting lesson learned about this synergy. When we have a correct notification, in only 30% cases the couriers change their behavior, But when we have an incorrect notification, for around 67% cases, the couriers will insist on their report and try to influence the system. That is, the couriers are more willing to change the system instead of changing their own behavior. And this leads to asymmetric synergy. There are two potential reasons for these results: Maybe the courier doesn't trust the platform [74] and just wants to ignore the notification because the platform gave an incorrect notification. Or the information asymmetry because as gig workers, they may not be aware of how the platform is operated. This can be an important topic, especially for human-AI collaboration in the future.

6.3 Order Scheduling

How to build a better system to help humans is the ultimate goal for almost all system research. In my research, we have worked on several interesting real-world applications to utilize the sensing and prediction results to make decisions to For example, we designed a public-transport-based crowdsourcing delivery scheme and formulate the order dispatching problem as an optimization problem to maximize the profit for the platform with guaranteed delivery time [205]. We also designed a Time-Constrained Actor-Critic Reinforcement learning-based concurrent dispatch system called TCAC-Dispatch to enhance the long-term overall revenue and reduce the overdue rate [206]. In this section, I will not discuss the technical details of the order dispatching algorithms since they are not directly related to the sensing and prediction results we have discussed in Chapter 4 and 5. Instead, I will use a simplified example to show how we incorporate the results from P^2-Loc discussed in Chapter 5 to better order dispatching.

Among all the applications of couriers' indoor localization (e.g., navigation for couriers, demonstration for merchants and customers, order scheduling for the platform), we show P^2-Loc's performance in order scheduling because the benefits can be directly measured by the monetary savings. We conduct an offline analysis on real-world *order and trajectory data* to compare the scheduling performance based on P^2-Loc GPS and Enc-MDS. It is estimated that the improved couriers' indoor locations can save \$ 40,000 for the platform every day nationwide.

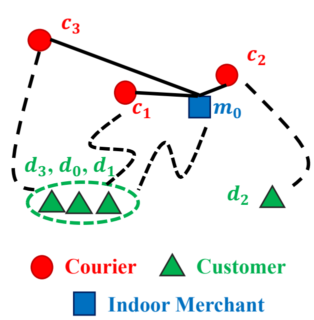


Figure 6.4: Order Scheduling

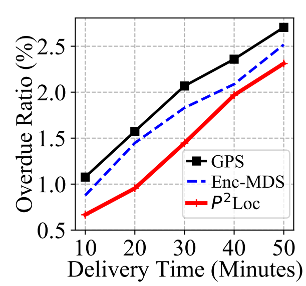


Figure 6.5: Application Results

Order Scheduling Background. A widely used scheduling strategy is that when an order is placed, it is assigned to a nearby courier with close destinations. As shown in Figure 6.4, when an order is placed in merchant m_0, the platform will check all the nearby couriers (i.e., c_1, c_2, and c_3). c_2 is the closest but c_2's destination d_2 is in the opposite direction. Both c_1's and c_3's destination (d_1 and d_3) are close to the destination of the new order (d_0). c_1 is closer to the merchant so the order will be assigned to c_1. This scheduling relies on the estimated distance between the merchants' GPS and couriers' GPS, which is inaccurate when indoor.

Analysis Settings. Because a badly dispatched order brings a terrible experience to both real couriers and customers, it is very challenging to conduct online *in-situ* experiments. Therefore, we obtain some real-world order scheduling data from the cooperating delivery platform, including the reporting timestamps, promised delivery time (different for each order), actual delivery time (from customers' feedback). The orders are considered as GPS-based. We select two subsets of the orders as `P`^{2}`-Loc`-based and Enc-MDS-based. Specifically, the orders that the same scheduling choice shall be made based on localization results provided by `P`^{2}`-Loc` and Enc-MDS-based method are considered as `P`^{2}`-Loc`-based and Enc-MDS-based. We use the overdue ratio (i.e., number of overdue orders among all orders) as a metric because it can (1) measure the performance of order scheduling, (2) measure the monetary benefit of `P`^{2}`-Loc` because the platform needs to compensate customers for each overdue order. The experiment is based on one month of orders in the mall area in Figure 5.18, including 33,915 orders and 3,701 couriers.

Results. From Figure 6.5 we observe that `P`^{2}`-Loc` reduces the overdue ratio by 0.5% compared with state-of-practice GPS method, and 0.3% compared with state-of-art encounter-based localization method. We think the reason that the seconds of improvement in travel time estimation can improve the whole delivery process is that there around 600 orders to be picked up by 100 riders nearby in the mall in the rush hour (11:00am 12:00pm), and the scheduling strategy is highly relied on estimated travel time of different courier-merchant pairs [66]. Therefore, even small improvement in travel time estimation will be amplified by the order scheduling process and impact on all the orders. Although the overdue reduction brought by `P`^{2}`-Loc` looks marginal (around 4‰), the total volume is enormous (more than 10 million daily orders of a single platform), and our method is infrastructure-free. Therefore it can be easily expanded nation-wide. Given that the platform covers $1 of the overdue penalty for each overdue order, it is estimated that \$40,000 can be saved for the platform every day based on $\texttt{P}^2\texttt{-Loc}$.

Chapter 7

Future Works

7.1 Overview

In the future, we will continue to work on the sensing and prediction of human behavior and mobility in human-centered cyber-physical systems. This goal requires a deeper understanding of human behavior, not only the mobility and activity that we have studied but also more behavior like their interaction with computers and other humans. This can be achieved on a wide range of sensors and communication technologies like Wi-Fi, Camera, IMU, and other new sensors in the future. Moreover, AI models and social science models will play key roles in building behavior models to predict human behavior, activities, locations, and mobility. To truly benefit the human, I will also work on a variety of actions like scheduling, notification, and recommendation in different applications. Like the beacon system we have built, I will not stop by just designing the systems, and I will try to implement and deploy the system not only in gig delivery but also in other systems like location-based social networks and intelligent transportation. In the following sections, I will discuss the future works in two directions.

7.2 Behavior and Mobility Study in Gig Delivery

The first direction is the couriers' behavior and mobility in gig delivery. For the behavior, the potential topics include the individual behavior like their reporting. And the group behavior like their interaction; We can use a computational social science perspective to study their behavior. For example, we can study how to build trust between gig workers and the platform. In the early arrival notification, we found that after some time, some couriers will directly choose to report now no matter whether he arrives or not. This may be due to a lack of trust in the platform because sometimes the platform gives incorrect notifications. Building trust between humans and systems is a critical problem for future human-AI collaboration applications. We can also use network science [207] to study problems like couriers' interaction. Figure 5.12 shows an interaction network of couriers and merchants in a mall for one day. It will be more complicated if we consider the merchants in the whole city for one month or even one year. We can study the power law and the hubs in this network to find the leaders or the experienced couriers in a group and their impact on other couriers.

For mobility, the first potential topic is the new challenges introduced by this new kind of mobility; the second potential topic is the economic and social observations from the massive gig delivery data. On one hand, we can study them as human mobility problems. Every day there are more than 1 million couriers working in different cities, delivering millions of orders from merchants to customers. All the couriers are riding electronic bikes on the roads that had been originally designed for cars or pedestrians. There are some problems we need to solve to let our transportation system work for this new mobility. For example, we can recognize couriers' dangerous driving behavior from their trajectories. Sometimes they may ride on the highway or the sidewalk, which is dangerous for themselves and pedestrians. On the other hand, we can also study mobility from the perspective of social science to get some insight into the economy or society. For example, we can predict the recovery or recession in the catering industry. It's important because it may impact how much money to invest and how many workers to recruit.

7.3 Behavior and Mobility Study in Smart City

The study of human behavior and mobility also has the potential to contribute the smart city applications. In this direction, there are several research topics that I am interested in. The first topic is the data cooperative in smart cities [208]. In the smart city setting, varieties of data related to human behavior and mobility will be collected by the government and service providers (e.g., network operators). Building a data cooperative will benefit many parties but also brings some challenges such as the tradeoff between privacy and data utility. There are many research problems to be explored such as the differences between moving data to the algorithms and moving algorithms to the data and how to measure this impact of privacy-utility trade-off in real-world applications. The second topic is human mobility in smart city applications. There are different types of mobility in the smart city setting. Escooter is one the emerging mobility in recent years, and there are several related research problems we can explore. For example, we can do the e-scooters rebalancing for fairness purposes by combining the escooter data with other data like demographic data.

Chapter 8

Conclusion

In this dissertation, we propose a three-layer human-centered cyber-physical system. For the sensing layer, we designed Bluetooth-based beacon systems to detect couriers' arrival at the merchants and solved the challenge of environment and smartphone device heterogeneity in real-world large-scale deployment. For the prediction layer, we infer couriers' indoor locations by predicting the travel time to nearby merchants based on the encounter events among couriers. For the decision-making, we use the example of early arrival notification to show how we utilize the feedback from couriers to improve the performance of the beacon system; and the example of order dispatching to show how to improve the dispatching performance based on the accurate indoor locations provided by the predicted travel time. All these works, although in different layers, share the same big vision, that is we want to build better systems for humans and build the synergy between humans and systems. The key factor in this big frame is human behavior, where we design different solutions with advanced communication technologies and learning techniques to sense them, predict them, and make decisions about them. With increasing interaction between humans and systems in the future, the ideas and solutions proposed in the dissertation can be applied to a variety of scenarios.

References

[1] Yi Ding, Ling Liu, Yu Yang, Yunhuai Liu, Desheng Zhang, and Tian He. From conception to retirement: a lifetime story of a 3-year-old wireless beacon system in the wild. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, Boston, MA, April 2021. USENIX Association.

[2] Yi Ding, Yu Yang, Wenchao Jiang, Yunhuai Liu, Tian He, and Desheng Zhang. Nationwide deployment and operation of a virtual arrival detection system in the wild. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 705–717, 2021.

[3] Wei Liu, Yi Ding, Shuai Wang, Yu Yang, and Desheng Zhang. Para-pred: Addressing heterogeneity for city-wide indoor status estimation in on-demand delivery. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3407–3417, 2022.

[4] Yi Ding, Dongzhe Jiang, Yu Yang, Yunhuai Liu, Tian He, and Desheng Zhang. P2-loc: A person-2-person indoor localization system in on-demand delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–24, 2022.

[5] Yi Ding, Yu Yang, Wenchao Jiang, Yunhuai Liu, Tian He, and Desheng Zhang. Nationwide deployment and operation of a virtual arrival detection system in the wild. *IEEE/ACM transactions on networking*, 2022.

[6] Christopher Greer, Martin Burns, David Wollman, Edward Griffor, et al. Cyber-physical systems and internet of things, 2019.

[7] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom*. ACM, 2012.

[8] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM conference on embedded networked sensor systems*, pages 1–14, 2013.

[9] Xiangyu Xu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Linghe Kong, and Minglu Li. Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals. In *ACM MobiSys*, pages 54–66, 2019.

[10] Bilong Shen, Xiaodan Liang, Yufeng Ouyang, Miaofeng Liu, Weimin Zheng, and Kathleen M Carley. Stepdeep: A novel spatial-temporal mobility event prediction framework based on deep neural network. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 724–733, 2018.

[11] Sijie Ruan, Zi Xiong, Cheng Long, Yiheng Chen, Jie Bao, Tianfu He, Ruiyuan Li, Shengnan Wu, Zhongyuan Jiang, and Yu Zheng. Doing in one go: Delivery time inference based on couriers' trajectories. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2813–2821, 2020.

[12] Arthur Delarue, Ross Anderson, and Christian Tjandraatmadja. Reinforcement learning with combinatorial actions: An application to vehicle routing. *Advances in Neural Information Processing Systems*, 33:609–620, 2020.

[13] Xiaocheng Tang, Zhiwei Qin, Fan Zhang, Zhaodong Wang, Zhe Xu, Yintai Ma, Hongtu Zhu, and Jieping Ye. A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1780–1790, 2019.

[14] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 1914–1933, 2017.

[15] Zheng Yao, Silas Weden, Lea Emerlyn, Haiyi Zhu, and Robert E Kraut. Together but alone: Atomization and peer support among gig workers. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–29, 2021.

[16] Zijian Zhang, Jaspreet Singh, Ujwal Gadiraju, and Avishek Anand. Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019.

[17] Fadel Adib and Dina Katabi. See through walls with wifi! In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 75–86, 2013.

[18] Patrick Dickinson, Gregorz Cielniak, Olivier Szymanezyk, and Mike Mannion. Indoor positioning of shoppers using a network of bluetooth low energy beacons. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2016.

[19] Lili Chen, Jie Xiong, Xiaojiang Chen, Sunghoon Ivan Lee, Kai Chen, Dianhe Han, Dingyi Fang, Zhanyong Tang, and Zheng Wang. Widesee: Towards wide-area contactless wireless sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 258–270, 2019.

[20] Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. Iodetector: A generic service for indoor outdoor detection. In *SenSys*. ACM, 2012.

[21] Kaikai Liu, Xinxin Liu, and Xiaolin Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 235–248, 2013.

[22] Shilin Zhu and Xinyu Zhang. Enabling high-precision visible light localization in today's buildings. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 96–108, 2017.

[23] Kartik Sankaran, Minhui Zhu, Xiang Fa Guo, Akkihebbal L Ananda, Mun Choon Chan, and Li-Shiuan Peh. Using mobile phone barometer for low-power transportation context detection. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 191–205, 2014.

[24] Pengfei Zhou, Mo Li, and Guobin Shen. Use it free: Instantly knowing your phone attitude. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 605–616, 2014.

[25] Yuanchao Shu, Cheng Bo, Guobin Shen, Chunshui Zhao, Liqun Li, and Feng Zhao. Magicol: Indoor localization using pervasive magnetic field and opportunistic wifi sensing. *IEEE Journal on Selected Areas in Communications*, 33(7):1443–1457, 2015.

[26] Jie Liu, Bodhi Priyantha, Ted Hart, Heitor S Ramos, Antonio AF Loureiro, and Qiang Wang. Energy efficient gps sensing with cloud offloading. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 85–98, 2012.

[27] Yinggang Yu, Dong Wang, Run Zhao, and Qian Zhang. Rfid based real-time recognition of ongoing gesture with adversarial learning. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 298–310, 2019.

[28] Ashutosh Dhekne, Ayon Chakraborty, Karthikeyan Sundaresan, and Sampath Rangarajan. {TrackIO}: Tracking first responders {Inside-Out}. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 751–764, 2019.

[29] John Nolan, Kun Qian, and Xinyu Zhang. Ros: passive smart surface for roadside-to-vehicle communication. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 165–178, 2021.

[30] Xuefeng Liu, Tianye Yang, Shaojie Tang, Peng Guo, and Jianwei Niu. From relative azimuth to absolute location: Pushing the limit of pir sensor based localization. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[31] Sinh Huynh, Rajesh Krishna Balan, JeongGil Ko, and Youngki Lee. Vitamon: measuring heart rate variability using smartphone front camera. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 1–14, 2019.

[32] Hua Huang, Hongkai Chen, and Shan Lin. Magtrack: Enabling safe driving monitoring with wearable magnetics. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 326–339, 2019.

[33] Justin Manweiler, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Predicting length of stay at wifi hotspots. In *2013 Proceedings IEEE INFOCOM*, pages 3102–3110. IEEE, 2013.

[34] Chao Chen, Qiang Liu, Xingchen Wang, Chengwu Liao, and Daqing Zhang. semitraj2graph: Identifying fine-grained driving style with gps trajectory data via multi-task learning. *IEEE Transactions on Big Data*, 2021.

[35] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*, pages 65–76, 2015.

[36] Ruiyang Gao, Mi Zhang, Jie Zhang, Yang Li, Enze Yi, Dan Wu, Leye Wang, and Daqing Zhang. Towards position-independent sensing for gesture recognition with wi-fi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–28, 2021.

[37] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. Using mobile phones to write in air. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 15–28, 2011.

[38] Sangwon Bae, Denzil Ferreira, Brian Suffoletto, Juan C Puyana, Ryan Kurtz, Tammy Chung, and Anind K Dey. Detecting drinking episodes in young adults using smartphone-based sensors. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 1(2):1–36, 2017.

[39] Jialin Liu, Dong Li, Lei Wang, and Jie Xiong. Blinklistener: " listen" to your eye blink using your smartphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–27, 2021.

[40] Mingmin Zhao, Fadel Adib, and Dina Katabi. Emotion recognition using wireless signals. In *Proceedings of the 22nd annual international conference on mobile computing and networking*, pages 95–108, 2016.

[41] Hanbin Zhang, Chen Song, Aosen Wang, Chenhan Xu, Dongmei Li, and Wenyao Xu. Pdvocal: Towards privacy-preserving parkinson's disease detection using non-speech body sounds. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[42] Abhirup Ghosh, Vaisakh Puthusseryppady, Dennis Chan, Cecilia Mascolo, and Michael Hornberger. Machine learning detects altered spatial navigation features in outdoor behaviour of alzheimer's disease patients. *Scientific Reports*, 12(1):1–13, 2022.

[43] Avik Ghose, Chirabrata Bhaumik, and Tapas Chakravarty. Blueeye: A system for proximity detection using bluetooth on mobile phones. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1135–1142, 2013.

[44] Chuhan Gao, Yilong Li, and Xinyu Zhang. Livetag: Sensing human-object interaction through passive chipless wi-fi tags. *GetMobile: Mobile Computing and Communications*, 22(3):32–35, 2019.

[45] Andreas Biri, Neal Jackson, Lothar Thiele, Pat Pannuto, and Prabal Dutta. Socitrack: Infrastructure-free interaction tracking through mobile sensor networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[46] Hiroyuki Satoh, Makoto Suzuki, Yukou Tahiro, and Hiroyuki Morikawa. Ambient sound-based proximity detection with smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, pages 1–2, 2013.

[47] Darshan Santani, Joan-Isaac Biel, Florian Labhart, Jasmine Truong, Sara Landolt, Emmanuel Kuntsche, and Daniel Gatica-Perez. The night is young: urban crowdsourcing of nightlife patterns. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 427–438, 2016.

[48] He Wang, Dimitrios Lymberopoulos, and Jie Liu. Local business ambience characterization through mobile audio sensing. In *Proceedings of the 23rd international conference on World wide web*, pages 293–304, 2014.

[49] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014, 2016.

[50] Yuecheng Rong, Zhimian Xu, Ruibo Yan, and Xu Ma. Du-parking: Spatio-temporal big data tells you realtime parking availability. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 646–654, 2018.

[51] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. Location prediction over sparse user mobility traces using rnns. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 2184–2190, 2020.

[52] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, Xinran Tong, and Hui Xiong. Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 305–313, 2019.

[53] Lu Duan, Yang Zhan, Haoyuan Hu, Yu Gong, Jiangwen Wei, Xiaodong Zhang, and Yinghui Xu. Efficiently solving the practical vehicle routing problem: A novel joint learning approach. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3054–3063, 2020.

[54] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In

Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 1720–1730, 2019.

[55] Tao-yang Fu and Wang-Chien Lee. Progrpgan: Progressive gan for route planning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 393–403, 2021.

[56] Yexin Li, Yu Zheng, and Qiang Yang. Efficient and effective express via contextual cooperative reinforcement learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 510–519, 2019.

[57] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. Cross-node federated graph neural network for spatio-temporal data modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1202–1211, 2021.

[58] Chengliang Gao, Fan Zhang, Yue Zhou, Ronggen Feng, Qiang Ru, Kaigui Bian, Renqing He, and Zhizhao Sun. Applying deep learning based probabilistic forecasting to food preparation time for on-demand delivery service. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2924–2934, 2022.

[59] Chengliang Gao, Fan Zhang, Guanqun Wu, Qiwan Hu, Qiang Ru, Jinghua Hao, Renqing He, and Zhizhao Sun. A deep learning method for route and time prediction in food delivery service. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2879–2889, 2021.

[60] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1695–1704, 2018.

[61] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand

forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3656–3663, 2019.

[62] Sijie Ruan, Cheng Long, Zhipeng Ma, Jie Bao, Tianfu He, Ruiyuan Li, Yiheng Chen, Shengnan Wu, and Yu Zheng. Service time prediction for delivery tasks via spatial meta-learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3829–3837, 2022.

[63] Sijie Ruan, Cheng Long, Jie Bao, Chunyang Li, Zisheng Yu, Ruiyuan Li, Yuxuan Liang, Tianfu He, and Yu Zheng. Learning to generate maps from trajectories. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 890–897, 2020.

[64] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. Representing urban functions through zone embedding with human mobility patterns. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018.

[65] Meng Qu, Hengshu Zhu, Junming Liu, Guannan Liu, and Hui Xiong. A cost-effective recommender system for taxi drivers. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 45–54, 2014.

[66] Zhe Xu, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, and Jieping Ye. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 905–913, 2018.

[67] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. Planning bike lanes based on sharing-bikes' trajectories. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1377–1386, 2017.

[68] Hao Liu, Yongxin Tong, Panpan Zhang, Xinjiang Lu, Jianguo Duan, and Hui Xiong. Hydra: A personalized and context-aware multi-modal transportation

recommendation system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2314–2324, 2019.

[69] Dhruv Agarwal, Srishti Agarwal, Vidur Singh, Rohita Kochupillai, Rosemary Pierce-Messick, Srinivasan Iyengar, and Mohit Jain. Understanding driver-passenger interactions in vehicular crowdsensing. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–24, 2021.

[70] Mashael Yousef Almoqbel and Donghee Yvette Wohn. Individual and collaborative behaviors of rideshare drivers in protecting their safety. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–21, 2019.

[71] Bhavani Seetharaman, Joyojeet Pal, and Julie Hui. Delivery work and the experience of social isolation. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–17, 2021.

[72] Keman Huang, Jinhui Yao, and Ming Yin. Understanding the skill provision in gig economy from a network perspective: A case study of fiverr. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019.

[73] Shaoyang Fan, Ujwal Gadiraju, Alessandro Checco, and Gianluca Demartini. Crowdco-op: Sharing risks and rewards in crowdsourcing. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–24, 2020.

[74] Eliscia Kinder, Mohammad Hossein Jarrahi, and Will Sutherland. Gig platforms, tensions, alliances and ecosystems: An actor-network perspective. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–26, 2019.

[75] Niels Van Berkel, Jorge Goncalves, Danula Hettiachchi, Senuri Wijenayake, Ryan M Kelly, and Vassilis Kostakos. Crowdsourcing perceptions of fair predictors for machine learning: A recidivism case study. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–21, 2019.

[76] Kapil Garg, Yongsung Kim, Darren Gergle, and Haoqi Zhang. 4x: A hybrid approach for scaffolding data collection and interest in low-effort participatory sensing. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–28, 2019.

[77] Liang Wang, Zhiwen Yu, Dingqi Yang, Tian Wang, En Wang, Bin Guo, and Daqing Zhang. Task execution quality maximization for mobile crowdsourcing in geo-social networks. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–29, 2021.

[78] Yi Wang. When didi is not really a choice in small chinese cities, taxi drivers build their own. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–30, 2019.

[79] Jason T Jacques and Per Ola Kristensson. Crowdworker economics in the gig economy. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2019.

[80] Amazon. Amzon prime now. Webpage, 2020.

[81] Uber. Uber eat. Webpage, 2020.

[82] InstaCart. Instacart. Webpage, 2020.

[83] DoorDash. Doordash. Webpage, 2020.

[84] Deliveroo. Deliveroo. Webpage, 2020.

[85] Meituan. Meituan. Webpage, 2020.

[86] Eleme. Eleme. Webpage, 2020.

[87] Lin Zhu, Wei Yu, Kairong Zhou, Xing Wang, Wenxing Feng, Pengyu Wang, Ning Chen, and Pei Lee. Order fulfillment cycle time estimation for on-demand food delivery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2571–2580, 2020.

[88] Junghyun Jun, Yu Gu, Long Cheng, Banghui Lu, Jun Sun, Ting Zhu, and Jianwei Niu. Social-loc: Improving indoor localization with social sensing. In *SenSys*. ACM, 2013.

[89] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. Indoor localization without the pain. In *MobiCom*. ACM, 2010.

[90] Andreas Haeberlen, Eliot Flannery, Andrew M Ladd, Algis Rudys, Dan S Wallach, and Lydia E Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom*. ACM, 2004.

[91] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *MobiCom*. ACM, 2014.

[92] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. Widar2. 0: Passive human tracking with a single wi-fi link. In *MobiSys*. ACM, 2018.

[93] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *ACM MobiCom*, pages 116–127, 2008.

[94] Zhao Tian, Yu-Lin Wei, Wei-Nin Chang, Xi Xiong, Changxi Zheng, Hsin-Mu Tsai, Kate Ching-Ju Lin, and Xia Zhou. Augmenting indoor inertial tracking with polarized light. In *ACM MobiSys*, pages 362–375, 2018.

[95] Yu-Lin Wei, Chang-Jung Huang, Hsin-Mu Tsai, and Kate Ching-Ju Lin. Celli: Indoor positioning using polarized sweeping light beams. In *MobiSys*. ACM, 2017.

[96] Bo Xie, Guang Tan, and Tian He. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *SenSys*. ACM, 2015.

[97] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via rf body reflections. In *NSDI'15 Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, pages 279–292, 2015.

[98] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. Are rfid sensing systems ready for the real world. In *ACM MobiSys*, pages 366–377, 2019.

[99] Shaoyi Zhu, Weiqing Huang, Chenggang Jia, Siye Wang, Bowen Li, and Yanfang Zhang. Rf-amoc: Human-related rfid tag movement identification in access management of carries. *ACM Transactions on Sensor Networks (TOSN)*, 16(4):1–23, 2020.

[100] Shahriar Nirjon, Robert F Dickerson, Philip Asare, Qiang Li, Dezhi Hong, John A Stankovic, Pan Hu, Guobin Shen, and Xiaofan Jiang. Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. In *ACM MobiSys*, pages 403–416, 2013.

[101] Qun Niu, Mingkuan Li, Suining He, Chengying Gao, S-H Gary Chan, and Xiaonan Luo. Resource-efficient and automated image-based indoor localization. *ACM Transactions on Sensor Networks (TOSN)*, 15(2):1–31, 2019.

[102] Jiayang Liu and Lin Zhong. Micro power management of active 802.11 interfaces. In *ACM MobiSys*, pages 146–159, 2008.

[103] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R. Smith. Passive wi-fi: Bringing low power to wi-fi transmissions. In *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, 2016.

[104] Ramsey Faragher and Robert Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.

[105] Daisuke Yamamoto, Ryosuke Tanaka, Shinsuke Kajioka, Hiroshi Matsuo, and Naohisa Takahashi. Global map matching using ble beacons for indoor route and stay estimation. In *ACM SIGSPATIAL*, pages 309–318, 2018.

[106] Yun Huang, Qunfang Wu, and Yaxing Yao. Bluetooth low energy (ble) beacons alone didn't work! In *ACM UbiComp*, pages 62–65, 2018.

[107] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. Accelprint: Imperfections of accelerometers make smartphones trackable. In *Network and Distributed System Security Symposium*, 2014.

[108] H. S. Ahn and W. Yu. Environmental-adaptive rssi-based indoor localization. *IEEE Transactions on Automation Science & Engineering*, 6(4):626–633, 2009.

[109] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Velickovic, and Pietro Lio. Using deep data augmentation training to address software and hardware

heterogeneities in wearable and smartphone sensing devices. In *IPSN*, pages 200–211, 2018.

[110] S. S. Shubha, T. Sen, H. Shen, and M. Normansell. A diverse noise-resilient dnn ensemble model on edge devices for time-series data. In *SECON*, pages 1–9, 2021.

[111] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. Crosssense: Towards cross-site and large-scale wifi sensing. In *MobiCom*, pages 305–320, 2018.

[112] Aditya Virmani and Muhammad Shahzad. Position and orientation agnostic gesture recognition using wifi. In *MobiSys*, pages 252–264, 2017.

[113] Yinggang Yu, Dong Wang, Run Zhao, and Qian Zhang. Rfid based real-time recognition of ongoing gesture with adversarial learning. In *SenSys*, pages 298–310, 2019.

[114] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, and Yaqing Wang. Towards environment independent device free human activity recognition. In *MobiCom*, pages 289–304, 2018.

[115] Alibaba Group. abeacon data-set. Webpage, 2021.

[116] Valid data-set link.
https://tianchi.aliyun.com/dataset/dataDetail?dataId=103969.

[117] Yi Ding, Dongzhe Jiang, Yunhuai Liu, Desheng Zhang, and Tian He. Smartloc: Indoor localization with smartphone anchors for on-demand delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–24, 2021.

[118] Dongzhe Jiang, Yi Ding, Hao Zhang, Yunhuai Liu, Tian He, Yu Yang, and Desheng Zhang. Alwaes: an automatic outdoor location-aware correction system for online delivery platforms. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–24, 2021.

[119] Pengfei Zhou, Yi Ding, Yang Li, Mo Li, Guobin Shen, and Tian He. Experience: Adopting indoor outdoor detection in on-demand food delivery business. 2022.

[120] Baoshen Guo, Weijian Zuo, Shuai Wang, Wenjun Lyu, Zhiqing Hong, Yi Ding, Tian He, and Desheng Zhang. Wepos: Weak-supervised indoor positioning with unlabeled wifi for on-demand delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–25, 2022.

[121] Google Developers. Services overview. Webpage, 2020.

[122] S. Yu, Q. Long, and Q. Yin. A c-lstm neural network for human activity recognition using wearables. In *ISSI*, 2018.

[123] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, pages 61–80, 2009.

[124] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[125] Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016, 1611.07308.

[126] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. In *IJCAI*, pages 3670–3676, 2019.

[127] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *CVPR*, pages 9211–9219, 2019.

[128] Heba Abdelnasser, Moustafa Youssef, and Khaled A. Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1472–1480, 2015.

[129] Markus Scholz, Till Riedel, Mario Hock, and Michael Beigl. Device-free and device-bound activity recognition using radio signal strength. In *Augmented Human International Conference*, pages 100–107, 2013.

[130] Daisuke Yamamoto, Ryosuke Tanaka, Shinsuke Kajioka, Hiroshi Matsuo, and Naohisa Takahashi. Global map matching using ble beacons for indoor route and stay estimation. In *SIGSPATIAL/GIS*, pages 309–318, 2018.

[131] Sam Shue and James M. Conrad. Reducing the effect of signal multipath fading in rssi-distance estimation using kalman filters. In *SpringSim (CNS)*, page 5, 2016.

[132] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, and Mads Mller Jensen. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *SenSys*, pages 127–140, 2015.

[133] Locatify. eldheimar museum, intuitively designed beacon based museum audio guide. `https://locatify.com/blog/case-studies/eldheimar-museum/`, 2020.

[134] Airport benchmarking. airlines and airports are beaconizing. `https://www.airportbenchmarking.com/airlines-and-airports-are-beaconizing/`, 2016.

[135] Government of Singapore. Tracetogether. , 2020.

[136] Yu Yang, Yi Ding, Dengpan Yuan, Guang Wang, Xiaoyang Xie, Yunhuai Liu, Tian He, and Desheng Zhang. Transloc: transparent indoor localization with uncertain human participation for instant delivery. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[137] Xuetao Ding, Runfeng Zhang, Zhen Mao, Ke Xing, Fangxiao Du, Xingyu Liu, Guoxing Wei, Feifan Yin, Renqing He, and Zhizhao Sun. Delivery scope: A new way of restaurant retrieval for on-demand food delivery service. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3026–3034, 2020.

[138] Yan Zhang, Yunhuai Liu, Genjian Li, Yi Ding, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. Route prediction for instant delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):124, 2019.

[139] DoorDash. How do scheduled orders work? Webpage, 2021.

[140] DoorDash. Where is my order? Webpage, 2021.

[141] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 299–314, 2010.

[142] Longfei Shangguan, Zheng Yang, Alex X Liu, Zimu Zhou, and Yunhao Liu. Relative localization of {RFID} tags using spatial-temporal phase profiling. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 251–263, 2015.

[143] Dong Li, Tarun Bansal, Zhixue Lu, and Prasun Sinha. Marvel: multiple antenna based relative vehicle localizer. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 245–256, 2012.

[144] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282, 2015.

[145] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *MobiCom*. ACM, 2015.

[146] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*, pages 376–388, 2018.

[147] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[148] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[149] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. Epsilon: A visible light based positioning system. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, pages 331–343, 2014.

[150] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. Are rfid sensing systems ready for the real world? In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 366–377, 2019.

[151] Sujay Narayana, R Venkatesha Prasad, Vijay S Rao, Tamma V Prabhakar, Sripad S Kowshik, and Madhuri Sheethala Iyer. Pir sensors: Characterization and novel localization technique. In *Proceedings of the 14th international conference on information processing in sensor networks*, pages 142–153, 2015.

[152] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14, 2007.

[153] Stephen P Tarzia, Peter A Dinda, Robert P Dick, and Gokhan Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *MobiSys*, pages 155–168. ACM, 2011.

[154] Chi Zhang and Xinyu Zhang. Litell: indoor localization using unmodified light fixtures. In *MobiCom*. ACM, 2016.

[155] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: Unsupervised indoor localization. In *MobiSys*. ACM, 2012.

[156] Chris Xiaoxuan Lu, Yang Li, Peijun Zhao, Changhao Chen, Linhai Xie, Hongkai Wen, Rui Tan, and Niki Trigoni. Simultaneous localization and mapping with power network electromagnetic field. In *MobiCom*. ACM, 2018.

[157] Matthias Grossglauser and Martin Vetterli. Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and*

Communications Societies (IEEE Cat. No. 03CH37428), volume 3, pages 1954–1964. IEEE, 2003.

[158] Henk Wymeersch, Jaime Lien, and Moe Z Win. Cooperative localization in wireless networks. *Proceedings of the IEEE*, 97(2):427–450, 2009.

[159] Ionut Constandache, Xuan Bao, Martin Azizyan, and Romit Roy Choudhury. Did you see bob? human localization using mobile phones. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 149–160, 2010.

[160] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 305–316, 2012.

[161] Heikki J Ailisto, Mikko Lindholm, Jani Mantyjarvi, Elena Vildjiounaite, and Satu-Marja Makela. Identifying people from gait pattern with accelerometers. In *Biometric Technology for Human Identification II*, volume 5779, pages 7–14. International Society for Optics and Photonics, 2005.

[162] Frauke Kreuter, Georg-Christoph Haas, Florian Keusch, Sebastian Bähr, and Mark Trappmann. Collecting survey and smartphone sensor data with an app: Opportunities and challenges around privacy and informed consent. *Social Science Computer Review*, 38(5):533–549, 2020.

[163] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[164] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803, 2019.

[165] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426, 2019.

[166] Alibaba Group, Local Service BU. P2loc data-set link. https://tianchi.aliyun.com/dataset/dataDetail?dataId=119097, 2022.

[167] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, 2004.

[168] Yi Shang and Wheeler Ruml. Improved mds-based localization. In *IEEE INFOCOM 2004*, volume 4, pages 2640–2651. IEEE, 2004.

[169] Andrew Symington and Niki Trigoni. Encounter based sensor tracking. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, pages 15–24, 2012.

[170] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. *ACM SIGCOMM Computer Communication Review*, 34(4):15–26, 2004.

[171] Ahmed A Ahmed, Hongchi Shi, and Yi Shang. Sharp: A new approach to relative localization in wireless sensor networks. In *25th IEEE International Conference on Distributed Computing Systems Workshops*, pages 892–898. IEEE, 2005.

[172] Will Hedgecock, Miklos Maroti, Janos Sallai, Peter Volgyesi, and Akos Ledeczi. High-accuracy differential tracking of low-cost gps receivers. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 221–234, 2013.

[173] Hyojeong Shin, Taiwoo Park, Seungwoo Kang, Bupjae Lee, Junehwa Song, Yohan Chon, and Hojung Cha. Cosmic: Designing a mobile crowd-sourced collaborative application to find a missing child in situ. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pages 389–398, 2014.

[174] Viet-Cuong Ta, Trung-Kien Dao, Dominique Vaufreydaz, and Eric Castelli. Smartphone-based user positioning in a multiple-user context with wi-fi and bluetooth. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 206–212. IEEE, 2018.

[175] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. Metasense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 110–123, 2019.

[176] Taegyeong Lee, Zhiqi Lin, Saumay Pushp, Caihua Li, Yunxin Liu, Youngki Lee, Fengyuan Xu, Chenren Xu, Lintao Zhang, and Junehwa Song. Occlumency: Privacy-preserving remote deep-learning inference using sgx. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–17, 2019.

[177] Chaoyun Zhang and Paul Patras. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 231–240, 2018.

[178] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. Mobility modeling and prediction in bike-sharing systems. In *Proceedings of the 14th annual international conference on mobile systems, applications, and services*, pages 165–178, 2016.

[179] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. Rf-echo: A non-line-of-sight indoor localization system using a low-power active rf reflector asic tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 222–234, 2017.

[180] Timothy J Pierson, Travis Peters, Ronald Peterson, and David Kotz. Proximity detection with single-antenna iot devices. In *ACM MobiCom*, pages 1–15, 2019.

[181] Nicholas D Lane, Petko Georgiev, and Lorena Qendro. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning.

In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 283–294, 2015.

[182] Apple. Apple developer forums. Webpage, 2020.

[183] Apple Inc. ibeacon. Webpage, 2020.

[184] Kleomenis Katevas, Hamed Haddadi, Laurissa Tokarchuk, and Richard G Clegg. Detecting group formations using ibeacon technology. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 742–752, 2016.

[185] Google. Get started with beacons. Webpage, 2021.

[186] Apple. Turning an ios device into an ibeacon device. Webpage, 2021.

[187] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[188] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

[189] Markus Jakobsson and Sid Stamm. Web camouflage: Protecting your clients from browser-sniffing attacks. *IEEE Security & Privacy*, 5(6):16–24, 2007.

[190] Jun Huang, Wahhab Albazrqaoe, and Guoliang Xing. Blueid: A practical system for bluetooth device identification. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 2849–2857. IEEE, 2014.

[191] Bahman Rashidi, Carol Fung, Anh Nguyen, Tam Vu, and Elisa Bertino. Android user privacy preserving through crowdsourcing. *IEEE Transactions on Information Forensics and Security*, 13(3):773–787, 2017.

[192] Arvin Wen Tsui Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, and Hao-Hua Chu. Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization. *IEEE Transactions on Mobile Computing*, 9(11):1551–1562, 2010.

[193] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[194] Wikipedia. Time-based one-time password. Webpage, 2021.

[195] Google. Monitor the battery level and charging state. Webpage, 2021.

[196] Guobin Shen, Zhuo Chen, Peichao Zhang, Thomas Moscibroda, and Yongguang Zhang. Walkie-markie: Indoor pathway mapping made easy. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pages 85–98, 2013.

[197] Google Developers. Wi-fi scanning overview. Webpage, 2020.

[198] Yifei Jiang, Yun Xiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Li Shang, and Michael Hannigan. Hallway based automatic indoor floorplan construction using room fingerprints. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 315–324, 2013.

[199] Google Cloud. Deploying models. Webpage, 2021.

[200] Yurong Xu, Yi Ouyang, Zhengyi Le, James Ford, and Fillia Makedon. Mobile anchor-free localization for wireless sensor networks. In *International Conference on Distributed Computing in Sensor Systems*, pages 96–109. Springer, 2007.

[201] Nissanka B Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 340–341, 2003.

[202] Kelen CT Vivaldini, Jorge PM Galdames, Thales S Bueno, Roberto C Araújo, Rafael M Sobral, Marcelo Becker, and Glauco AP Caurin. Robotic forklifts for intelligent warehouses: Routing, path planning, and auto-localization. In *2010 IEEE International Conference on Industrial Technology*, pages 1463–1468. IEEE, 2010.

[203] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pages 1–6, 2020.

[204] Youxuan Lai. Couriers are trapped in the systems. Webpage, 2021.

[205] Yi Ding, Baoshen Guo, Lin Zheng, Mingming Lu, Desheng Zhang, Shuai Wang, Sang Hyuk Son, and Tian He. A city-wide crowdsourcing delivery system with reinforcement learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–22, 2021.

[206] Baoshen Guo, Shuai Wang, Yi Ding, Guang Wang, Suining He, Desheng Zhang, and Tian He. Concurrent order dispatch for instant delivery with time-constrained actor-critic reinforcement learning. In *2021 IEEE Real-Time Systems Symposium (RTSS)*, pages 176–187. IEEE, 2021.

[207] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

[208] Thomas Hardjono and Alex Pentland. Data cooperatives: Towards a foundation for decentralized personal data management. *arXiv preprint arXiv:1905.08819*, 2019.

Appendix A

Published Work

In addition to this dissertation, my research results are also given by the following published work.

[SIGSPATIAL'22] Zhiqing Hong, Guang Wang, Wenjun Lyu, Baoshen Guo, **Yi Ding**, Haotian Wang, Shuai Wang, Yunhuai Liu, Desheng Zhang. CoMiner: Nationwide Behavior-driven Unsupervised Spatial Coordinate Mining from Uncertain Delivery Events. In ACM SIGSPATIAL 2022.

[KDD'22] Wei Liu*, **Yi Ding***, Shuai Wang, Yu Yang, and Desheng Zhang. Para-Pred: Addressing Heterogeneity for City-Wide Indoor Status Estimation in On-Demand Delivery. In ACM SIGKDD 2022. 195/753=26% (*= equal contribution)

[IMWUT/UbiComp'22] Baoshen Guo, Weijian Zuo, Shuai Wang, Wenjun Lyu, Zhiqing Hong, **Yi Ding**, Tian He, Desheng Zhang. WePos: Weak-supervised Indoor Positioning with Unlabeled WiFi for On-demand Delivery. In ACM UbiComp 2022.

[MobiCom'22] Pengfei Zhou, **Yi Ding**, Yang Li, Mo Li, Guobin Shen, and Tian He. Experience: Adopting Indoor Outdoor Detection in On-demand Food Delivery Business. In ACM MobiCom'22.

[IMWUT/UbiComp'22] **Yi Ding**, Dongzhe Jiang, Yu Yang, Yunhuai Liu, Tian He, and Desheng Zhang. P2-Loc: A Person-2-Person Indoor Localization System in On-Demand

Delivery. In ACM UbiComp 2022.

[IMWUT/UbiComp'22] **Yi Ding**, Dongzhe Jiang, Yunhuai Liu, Desheng Zhang, and Tian He. SmartLOC: Indoor Localization with Smartphone Anchors for On-Demand Delivery. In ACM UbiComp 2022.

[RTSS'21] Baoshen Guo, Shuai Wang, **Yi Ding**, Guang Wang, Suining He, Desheng Zhang, and Tian He. Concurrent Order Dispatch for Instant Delivery with Time-Constrained Actor-Critic Reinforcement Learning. In IEEE RTSS 2022. *Outstanding Paper Award.*

[IMWUT/UbiComp'21] Dongzhe Jiang, **Yi Ding**, Hao Zhang, Yunhuai Liu, Tian He, Yu Yang, and Desheng Zhang. ALWAES: an Automatic Outdoor Location-Aware Correction System for Online Delivery Platforms. In ACM UbiComp 2021.

[IMWUT/UbiComp'21] **Yi Ding**, Baoshen Guo, Lin Zheng, Mingming Lu, Desheng Zhang, Shuai Wang, Sang Hyuk Son, and Tian He. A City-Wide Crowdsourcing Delivery System with Reinforcement Learning. In ACM UbiComp 2021.

[SIGCOMM'21] **Yi Ding**, Yu Yang, Wenchao Jiang, Yunhuai Liu, Tian He, Desheng Zhang. Nationwide Deployment and Operation of a Virtual Arrival Detection System in the Wild In ACM SIGCOMM 2021. 55/241=23%

[NSDI'21] **Yi Ding**, Ling Liu, Yu Yang, Yunhuai Liu, Tian He, Desheng Zhang. From Conception to Retirement: a Lifetime Story of a 3-Year-Old Operational Wireless Beacon System in the Wild. In USENIX NSDI 2021. 19/114=16%.

[MobiCom'20] Yu Yang, **Yi Ding**, Dengpan.Yuan, Guang Wang, Xiaoyang Xie, Yunhuai. Liu, Tian He, and Desheng Zhang. TransLoc: Transparent Indoor Localization with Uncertain Human Participation. In ACM MobiCom'20. 63/384=16%.

[IMWUT/UbiComp'19]Yan Zhang, Yunhuai Liu, **Yi Ding**, Genjian Li, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. Route Prediction for Instant Delivery. In ACM UbiComp 2019.

Appendix B

Datasets Release

- aBeacon: collected from a citywide physical beacon system, including the data of Bluetooth, manual report, and location from 31,131 couriers at 2,466 merchant locations in one month. [Link]

- VALID: collected from a nationwide virtual beacon system, including the data of Bluetooth, manual report, and manual feedback from 55,000 couriers at 113,000 merchant locations in ten cities in one month. [Link]

- RL-Dispatch: collected from a citywide crowdsourcing delivery system, including the on-demand delivery order data in one month. [Link]

- ALWAES: collected in a location correction project, including the statistics of order update data, route network information from 20,000 couriers at 3,000 merchant locations in one city in one month. [Link]

- P2Loc: collected in a encounter-based localization project, including the BLE-based indoor encounter data and reported data of 176 couriers at 54 merchant locations in a mall in one month.. [Link]