TransLoc: Transparent Indoor Localization with Uncertain Human Participation for Instant Delivery

Yu Yang[§†], Yi Ding[§‡], Dengpan Yuan[†], Guang Wang[†], Xiaoyang Xie[†]
Yunhuai Liu[⋆], Tian He[§], Desheng Zhang[†]
[§]Alibaba Group, Local Services BU, [†]Rutgers University, [‡]University of Minnesota, [⋆]Peking University

ABSTRACT

Instant delivery is an important urban service in recent years because of the increasing demand. An important issue for delivery platforms is to keep updating the status of couriers especially the real-time locations, which is challenging when they are in an indoor environment. We argue the previous indoor localization techniques cannot be applied in the instant delivery scenario because they require extra deployed infrastructures and extensive labor work. In this work, we perform the couriers' indoor localization transparently in a predictive manner without extra actions of couriers by existing data from the platform including order progress reports and couriers' trajectories. Specifically, we present *TransLoc* to predict couriers' indoor locations by addressing two challenges including uncertain reporting behaviors and uncertain indoor mobility behaviors. Our key idea lies in two insights (i) couriers' behaviors are consistent in indoor/outdoor environments; (ii) localization, as a spatial inference problem, could be converted to a temporal inference problem. We evaluate *TransLoc* on 565 couriers from an instant delivery company, which improves baselines by at most 72%, and achieves a competitive result compared to a label-extensive approach. As a case study, we apply *TransLoc* to optimize the order dispatching strategy, which reduces the delivery time by 24%.

CCS CONCEPTS

• **Information systems** → **Mobile information processing systems**; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools.*

KEYWORDS

Instant delivery, courier behaviors, indoor localization

ACM Reference Format:
Yu Yang[§†], Yi Ding[§‡], Dengpan Yuan[†], Guang Wang[†], Xiaoyang Xie[†] and Yunhuai Liu[⋆], Tian He[§], Desheng Zhang[†]. 2020. TransLoc: Transparent Indoor Localization with Uncertain Human Participation for Instant Delivery. In *The 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20), September 21–25, 2020, London, United Kingdom.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3372224.3419198

MobiCom '20, September 21–25, 2020, London, United Kingdom

ACM ISBN 978-1-4503-7085-1/20/09…$15.00
https://doi.org/10.1145/3372224.3419198

1 INTRODUCTION

Instant delivery is an increasingly important urban service in the recent several years driven by the increasing demand for quick product delivery [48], especially in the background of the coronavirus outbreak [41]. Compared with traditional delivery services such as FedEx requiring days, instant delivery is an extremely fast delivery service (e.g., 30 mins for food or 1 hour for grocery) conducted by platforms such as PrimerNow [1], UberEats [47], DoorDash [15], Postmates [34], Instacart [22], MeiTuan [32], and Eleme [16]. For instant delivery, one of the most important factors is real-time locations of couriers, which are the key for delivery order dispatching under its extremely short deadline [10] [27] [43]. If an order was delivered after the deadline, the delivery service provider may have to pay an overdue fee to a customer. Different from outdoor locations (i.e., GPS) that can be collected from couriers' work smartphones, large-scale indoor locations are typically not available but are essential for real-time delivery dispatching in multi-floor buildings in big cities, e.g., New York and Shanghai. We found we can save 7 minutes on average (i.e., counting for 25.9% of the delivery time) by simply modifying the current dispatching strategy if we know couriers' indoor locations, which reduces the overdue rate by 2.5% (as shown in § 7.6). This not only helps improve user experience but also saves overdue fees for the platforms considering totally more than 12 million daily orders.

To date, the indoor localization problem has been extensively studied with techniques such as fingerprinting (e.g., WiFi ID [11]), wireless signal modeling (e.g., RSSI [12], time of flight [2] and angle of arrival [25]), and models based on smartphone inertial sensors [50] (e.g., accelerometers, gyroscopes, magnetometers). However, compared with them, we emphasize two unique *design goals* in the instant delivery localization solution: *(i) No Additional Infrastructure/Label Investment:* the solution should not require any additional infrastructure investment (e.g., ultra-bandwidth devices [29]) or extensive labeling efforts (e.g., manually collected indoor fingerprinting) for large-scale low-cost deployment. *(ii) No Additional Human Input:* the solution should not require additional participating activities from couriers, i.e., the couriers just perform regular delivery and reporting on the smartphones. We name localization under these requirements as **transparent localization** that is achieved without any extra efforts from couriers or new infrastructures.

We explore two new opportunities to achieve transparent localization for instant delivery. **(1) Order Progress Reporting:** After a platform assigns an order to a courier, it is mandatory for a courier to manually report the order progress in 4 major stages (shown in Fig. 1) on her/his smartphone including accepting an order (t_0), arriving at a merchant (t_1), picking up the order (t_2), and the order delivered (t_3), to inform the customer and platform the

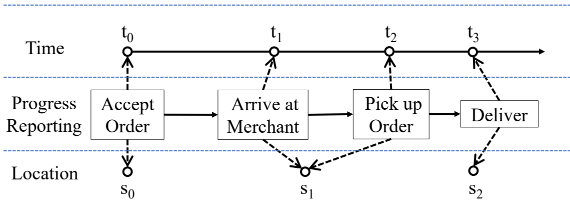


Fig 1: Order Progress Reporting

real-time order status for a better experience under short delivery time. When couriers head to indoor merchants, we have two important temporal anchors (i.e., t_1 and t_2) to provide the context of a courier's indoor status. **(2) Logical Localization Accuracy:** Instead of centimeter-level localization [54] [49] [2] that are expensive to achieve in large-scale settings, we aim to design a cheaper localization approach while still provide sufficient location information for scheduling. We found, for instant delivery platforms, the key factor to determine the courier scheduling is the worker's logical location (i.e., which merchant a courier is closest to). In this case, the localization granularity of couriers could potentially be relaxed to the logical accuracy on the merchant level, instead of physical accuracy. Based on these two opportunities, our research question is *can we utilize order progress reporting to localize couriers on the merchant level when they are indoor?*

To answer this question, we perform a case study in the Alibaba's instant delivery platform in Chinese city Shanghai. Based on a real-world field study, we found two key challenges when we explore these two opportunities. **(1) Uncertain Reporting Behaviors:** The manual order progress reporting is extremely unreliable with a significant number of early or late progress reports, especially in the arrive-at-merchant stage (i.e., t_1) due to the overdue penalty. Based on real-world data, we found 55.2% of orders have early arrival reporting and 1.6% of orders have late arrival reporting longer than 1 minute, which makes it challenging to associate these timestamps to correct locations. **(2) Uncertain Indoor Mobility Behaviors:** Reporting behaviors only provide sparse information regarding the pickup merchant without continuous traces (e.g., other nearby passing merchants on the way), which cannot be used to localize couriers in real time before arriving at pickup merchants. In addition, the indoor maps may not be available for a large scale deployment, which makes it even more challenging. We will show the detailed analysis of these two challenges in § 2.3.

To address them, we design a prototype system called *TransLoc* to utilize order progress data (i.e., three anchors per order with uncertain location-time contexts) to localize couriers on the merchant level. Our system is based on two key insights: (1) we found most couriers' outdoor/indoor reporting behaviors are intrinsically consistent under certain context, which could be used to address the uncertain reporting behaviors; (2) the original spatial inference problem (i.e., logistical localization) could be converted to a temporal inference problem (i.e., walking time inference), based on which we could predict when a courier arrives at each merchant and achieve localization in a predictive manner.

Considering the real-world constraints for large-scale commercial platforms, instead of deploying expensive infrastructures (e.g., Wi-Fi maps or RSSI mapping), our work is to address a classic mobile system problem in a data-driven and more scalable way. We demonstrate the power of in-depth analytics and reuse on already collected data, which is useful for improving the service quality to customers, workers, and suppliers. By incorporating human behavior modeling (i.e., reporting behaviors) in the system design, we achieve localization without extra infrastructure support. Further, we convert the classic spatial localization problem that requires real-time signal detection (e.g., Wi-Fi and Bluetooth) into a temporal prediction problem (e.g., predicting the arrival time), which is more feasible to be addressed in the delivery platform. As for the detailed technique, we focused on its robustness and simplicity to make it applicable in a large-scale deployment. Specifically, the key contributions of this paper are given as follows.

- To our knowledge, we performed the first study of transparent indoor localization in an instant delivery platform built on the existing infrastructure with low overhead. We design *TransLoc* based on a real-world setting of one week with 565 couriers, 128 merchants, and 14,743 orders. While *TransLoc* is a novel system for instant delivery, we believe that our broader contribution is in revealing a new direction in addressing the traditional indoor localization problem. Our key advantages are high penetration (i.e., all workers voluntarily report, which means we get reporting for free with consent); high applicability (i.e., neither new deployment nor extra worker participation needed). Thus, its principle can be easily generalized to other environments with two conditions: (1) users report their status under a context; (2) users' mobility is rather logically linear (e.g., from one place to another). We will share our data for the benefits of the community.
- To address the reporting uncertainty, based on the reporting behavior consistency, we first design a base model with common behaviors regardless of indoor/outdoor environments. Then we adapt it to the indoor environment using a transfer learning technique with a newly introduced constraint and data samples. To address the indoor mobility uncertainty, we design a symbolic mobility graph based on indoor walking time, which converts the localization problem into a walking time inference problem.
- We evaluate our system by the ground truth collected from deployed Bluetooth Beacon devices in 27 indoor merchants, which provides locations and time when couriers are nearby. Experiments show *TransLoc* improves the localization accuracy by 64% and 72% compared with Wi-Fi/GPS-based methods, and has a competitive accuracy with a label-extensive fingerprinting-based method. To quantify the benefit of our *TransLoc*, we conduct a field study of optimizing order dispatching strategies given predicted courier's indoor locations. The results show we reduce the courier walking time by 141 seconds and lead to a 24% improvement compared with the current dispatching.

2 BACKGROUND AND MOTIVATION

2.1 Data

2.1.1 ***Instant Delivery Order Progress.*** An order progress record logs all the information since a customer places an order until the order is successfully delivered. We only list the fields we use in this paper in Table 1. Note that these timestamps are uploaded in real time by courier smartphone apps, so we are aware of the real-time status of couriers. In our study, we collected data of 14,743 orders involving 565 couriers and 128 merchants in one week.

Table 1: Order Progress Record Format and Example

Field	Value
Order/Courier/Merchant ID	O001/C001/R001
Merchant Location	31.231728, 121.380751
Accepting Order Time	01/01/2019 12:05:00
Reported Arrival Time	01/01/2019 12:16:00
Reported Pickup Time	01/01/2019 12:16:10
Reported Delivered Time	01/01/2019 12:31:00

Table 2: Coordinate Types

Type	Detail	Type	Detail
0	Not Available	4	Cache
1	GPS Module	5	Wi-Fi
2	Last Coordinate	6	Cellular Tower

To facilitate the discussion, we categorize merchants into two types: **outside-merchant** and **inside-merchant**, depending on whether it is located inside a building. An outside-merchant means people get into the merchant from an open area such as a merchant on the roadside; an inside-merchant means the merchant is located in an indoor environment such as a multi-floor building.

2.1.2 ***Courier Trajectories.*** Couriers' trajectories contain continuous location information when couriers are in working, which are obtained from APIs of an online map service [21] deployed in couriers' smartphones. The location information is uploaded to the platform including coordinates, timestamps, and speeds in a frequency of 20 seconds under courier consents. In addition, provided by the online map service, each coordinate is assigned with a specific type, indicating how the coordinate is collected. We list six types in Table 2 (there is no type 3 from the service API) and two most common types are type 1 and type 5, which indicate if the coordinates are directly from a GPS module (i.e., type 1) or approximated locations from a WiFi localization method (i.e., type 5) when the GPS signal is not strong. Note that indicated by the service provider, the type 5 WiFi localization has errors ranging from 5 to 200 meters because of its mechanism of collecting the locations of Wi-Fi access points [21]. The basic mechanism behind is crowdsourcing GPS coordinates of nearby smartphones that can observe the access points, which could lead to dozens of meters away considering the long-range Wi-Fi signals [6]. All the data are obtained legally under the couriers' consent.

2.2 Problem Setting for Localization

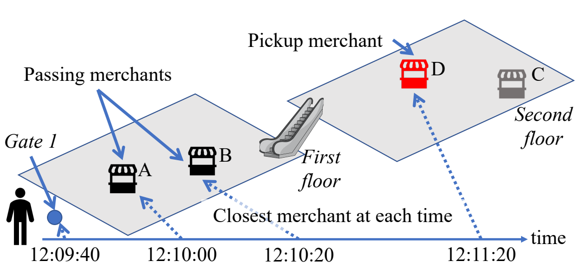


Fig 2: Problem Setting Demonstration

Given the historical delivery order progress and trajectories of couriers, we model their reporting behaviors (i.e., obtain the correct historical arrival time) and learn their historical indoor mobility models (i.e., estimate the walking time to each merchant). In real-time localization, as shown in Fig. 2, our solution works as two steps: *detection* and *prediction*. (i) Detection: we first detect when (e.g., 12:09:40) and where (e.g., Gate 1) a courier enters a building based on the above trajectory data uploaded every 20 seconds. (ii) Prediction: we then predict which merchant in a multi-floor building is closest to the courier continuously (e.g., every 10 seconds) until (s)he arrives at the pickup merchant. For example, *A* is the closest merchant to the courier at 12:10:10. Note that not all the merchants are passed such as *C*. Given these logical level locations, the platform can dispatch real-time pickup orders to potential couriers, e.g., the closest courier.

In our problem, different from the classic localization problem that localizes people or devices based on real-time signals such as Wi-Fi and Bluetooth, we achieve localization in a predictive manner by predicting couriers' arrival time to different locations (i.e., merchants), which tells us where couriers are at a specific time. Based on the predicted arrival time and their visited path, we predict couriers' locations at any given time after entering the building, which are the same outputs of the classic localization problems so we consider it a localization problem in our work.

2.3 Two Challenges

(i) Uncertain Reporting Behaviors: We analyze the courier's reporting behaviors by comparing the reported arrival time and the actual arrival time from the ground truth (details in § 7). Fig. 3a plots the proportion of the difference between the actual arrival time and reported arrival time. We name the difference as *reporting error*. It shows that (1) only 28.6% of the orders have the reporting error within 1 minute; (2) about 55.2% of the orders have issues of early reporting more than 1 minute; (3) about 19.6% of the orders even differ by more than 10 minutes. The reason is that due to the time-sensitive nature of instant delivery, the platform takes a strict policy to penalize a late (i.e., overdue) delivery. Either merchants or couriers should take responsibility depending on the late delivery is because of late preparation or late pickup. Some couriers may report the progress earlier than the real progress to avoid responsibility if the delivery is finally overdue. Sometimes couriers may also forget to report progress that leads to very late reporting. As a result, the uncertainty of couriers' reporting behaviors brings significant challenges to our problem.

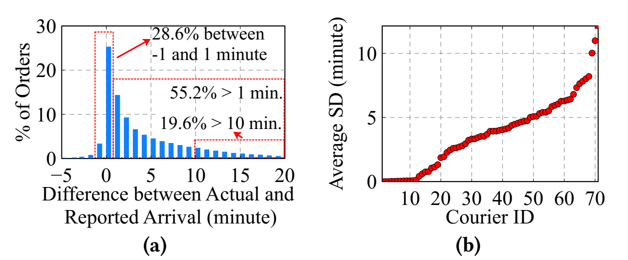
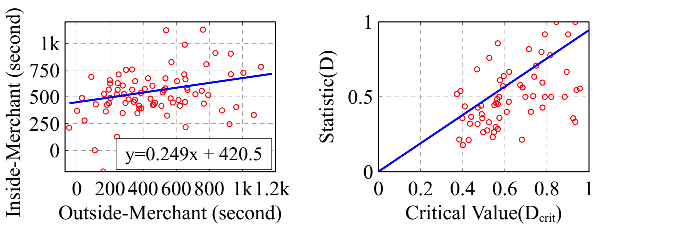


Fig 3: (a) The difference between actual arrival time and reported arrival time; (b) The average standard deviation (SD) of walking time for each courier.

(ii) Uncertain Indoor Locations: If we can correct the uncertain reporting, a straightforward way to localize couriers is based on their average walking speed and dead-reckoning on the map [50]. However, we found courier's indoor mobility may be affected by many factors such as routes, walking speeds, order assigned, and elevators. We show the uncertainty by studying the couriers' walking time from the same entrance of a multi-floor building to merchants. Fig. 3b plots the average walking time Standard Deviation (SD) between the same entrance and merchant for 70 couriers in our tested mall. On average, couriers have a standard deviation of 226 seconds (around 3.8 minutes), which shows the great variance to infer their indoor locations under different contexts. In addition, indoor environment information such as indoor maps is not always available compared with the outdoor. How to construct indoor maps efficiently on a large scale is still an open problem [19]. In our platform, it involves thousands of malls/buildings in different cities, which are expensive and may not be realistic to obtain all the indoor maps. To make our system more generic and practical, we do not assume the availability of the indoor maps, which introduces an extra challenge to localize couriers.

2.4 Two Key Ideas

We address these challenges by two key ideas:
(i) Consistent Courier Behavior for Outside-merchants and Inside-merchants: Our goal is to localize couriers in the indoor environment via courier reporting behavior modeling. Compared with previous human modeling work with available labels [18] [20], it is difficult to obtain the actual arrival time at the inside-merchants based on the existing platform considering the inaccurate indoor GPS signal. We argue that compared with inaccurate inside-merchant arrival time, it is relatively easy and confident to obtain the arrival time in an outside-merchant because we know when a courier arrives at the outside-merchant by their trajectories. If the courier's reporting behavior is consistent regardless (s)he heads to an inside/outside-merchant, the model formulated for the outside-merchant reporting behavior also has the potential to represent her/his reporting behavior at inside-merchants. To verify this intuition, we analyze the correlation of reporting errors between outside-merchants and inside-merchants.



(a) Mean Reporting Error **(b) Hypothesis Test**

Fig 4: (a) The average reporting error of couriers at the outside/inside-merchants; (b) Kolmogorov-Smirnov test on the outside/inside-merchants reporting error.

We first plot the mean reporting error with a fitting line in Fig. 4a, where each point represents a courier. We observed a positive correlation for most couriers, i.e., if a courier has larger reporting errors at outside-merchants, (s)he generally has larger reporting errors at inside-merchants. To quantify the correlation statistically, we conduct a Kolmogorov-Smirnov hypothesis test on individual couriers' behaviors under level 0.05. The null hypothesis is that their behaviors are from the same distribution regardless of outside-merchants or inside-merchants. We compute two values in the test: Kolmogorov-Smirnov statistic (D) and critical value (D_{crit}). If $D < D_{\text{crit}}$, the null hypothesis is accepted. Fig. 4b plots the computed values of each courier, and the diagonal line represents $D = D_{\text{crit}}$. We found there are 75.8% of the couriers are below the line, which means they statistically have consistent reporting behaviors. Given these two observations in Fig. 4a and 4b, we argue that most couriers perform consistent reporting behaviors, which provides the statistical foundation for a unified reporting model. Indeed, Fig. 4a does not show a strong correlation. However, from the hypothesis test result, we found our hypothesis of consistent outside/inside-merchants reporting cannot be rejected for most of the couriers. More importantly, Fig. 4a serves as a qualitative result, which motivates us to study the reporting consistency. Our evaluation quantitatively validates that the reporting consistency can result in a good performance on the arrival time estimation.

(ii) Convert spatial inference to temporal inference: As aforementioned in § 1, our work aimed to achieve logical merchant-level localization considering the required accuracy of instant delivery. Instead of solving it as spatial inference, we convert it into a temporal inference. The intuition is that if we know the walking time to each merchant from a certain location (e.g., gate) and the visiting sequence, we could potentially infer which merchant is nearest to a courier at any given time after passing the gate. To this end, we construct a symbolic mobility graph where each node represents a merchant; each edge represents the connection (i.e., path) between two merchants. Then the weight of edges represents the walking time between nodes, in which way we can find the closest graph node to a courier in terms of their walking time. Compared with physical localization, an important benefit of a symbolic mobility graph is that the edge length is directly used to measure the closeness between couriers and merchants in terms of the actual walking time, while the physical distance may not be propositional to the walking time considering the complex indoor environment. In addition, comparing other physical partitions, the symbolic mobility graph also helps us reduce computational complexity. For example, one of the largest shopping malls in Shanghai has 57 merchants(nodes) covering more than 40 thousand square meters, which leads to 10 thousand nodes if dividing the indoor environment into equal-sized grids (e.g., 2 by 2 meters [56]).

3 OVERVIEW

We present the overview of our system design in Fig. 5 including three modules.
Delivery Platform: The platform contains all the functionality of the existing instant delivery platform. We simplify it to three components used in our work including (1) the trajectory repository, (2) the order progress repository, and (3) the real-time trajectory and order progress.
Reporting Module (§ 4): In this module, we model the reporting behaviors of couriers to obtain the corrected arrival time. We first design a base model based on common features in both inside/outside-merchants. Then we modify and adapt it to model the inside-merchant reporting behaviors with unique observations for inside-merchants, which finally outputs the corrected indoor arrival time.

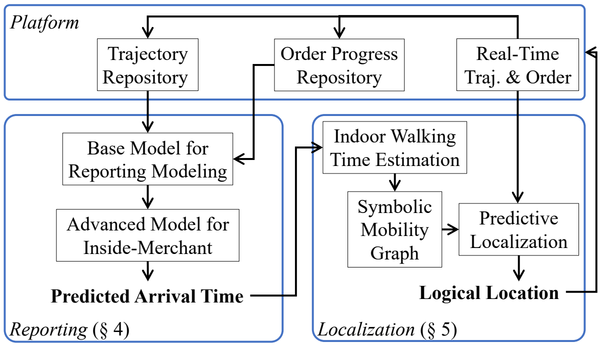


Fig 5: Design Overview

Localization Module (§ 5): Given the corrected arrival time in the reporting module, we formulate the indoor walking time into two-dimensional matrices where each entry represents the walking time from an entrance to a merchant or from a merchant to another merchant. Based on the walking time matrices, we construct a symbolic mobility graph, which represents the visiting order and time between merchants. In real time, given the trajectory and the order progress of a courier, we search the most likely indoor mobility path in the graph, which represents the logical indoor location at any given time. Finally, these locations are feedback to the platform to optimize order dispatching and generate new trajectories and order progress.

4 BEHAVIOR MODELING FOR ARRIVAL TIME PREDICTION AT MERCHANTS

In this section, we introduce how we model couriers' reporting behaviors by a base model and an advanced model.

4.1 Base Model for Arrival Time Prediction

4.1.1 ***Feature Extraction for Arrival Time Prediction.*** We show several features from two aspects: real-time features and historical features. Note that these features are common for both outside-merchants and inside-merchants. For an order, the real-time features of a courier include:

- Accepting Time (AT) represents the time when a courier accepts an order.
- Relative Reporting Time (RRT) is the time between the accepting time to the reported arrival time (t_1 in Fig.1).
- Distance to Merchant (DM) is the distance to the merchant at the reported arrival time.
- Concurrent Orders (CO): A courier may have multiple concurrent orders from nearby merchants. This feature shows the number of unfinished orders when a courier reports his/her status.
- Time Budget (TB): To ensure the on-time delivery, the platform generally sets a time constraint for delivery deadline based on the empirical study such as 30 minutes. When the time constraint is approaching, couriers may report ahead of time. We define the time budget as the remaining time to the delivery deadline when the courier reports his/her status, e.g., arrival at a merchant.

The historical features of a courier include:

- Number of Historical Orders of Pickup merchant (NHOP): It represents the delivery experience of a courier from a specific merchant.
- Number of Historical Orders of a courier (NHO): It represents the overall experience of a courier.
- Historical Average reporting Error of Pickup merchant (HAEP): the average reporting error of historical orders of a courier at a specific merchant.
- Historical Average reporting Error of All merchants (HAEA): the average reporting error of historical orders of a courier at all merchants.

Features Importance: We further study the importance of these features based on the random forest algorithm. The feature importance is obtained based on the inherent feature of Random Forests using impurity based ranking, which is a common approach used in data mining [39]. The result is: HAEP(0.236) > RRT(0.176) > DM(0.167) > CO(0.102) > NHO(0.093) > HAEA(0.084) > TB(0.078) > NHOP(0.048) > AT(0.016). The larger the value is, the higher importance the feature is. A few interesting observations are implied by the feature importance such as (1) HAEP with the highest importance implies couriers generally have stable reporting behaviors on the same merchant; (2) AT with the lowest importance implies the time of a day has a low impact on couriers' behaviors.

4.1.2 ***Courier Grouping.*** In an ideal case, we should design a prediction model for each courier, but the training data may not be sufficient at an individual level and it could also lead to overfitting. So, we classify couriers into groups based on their actual and reported arrival time. As in Fig. 6, the reporting error of couriers varies a lot, i.e., 12.9% of couriers have errors less than 60 seconds and 22.3% of them are greater than 360 seconds. The principle of grouping is that couriers in the same group should have similar reporting behaviors (i.e., reporting errors) at the same merchants. Formally, this principle can be presented by *cscore* defined as

$$cscore(c_i, c_j) = \sum_{m \in M} KS(S_i^m, S_j^m) \quad (1)$$

where c_i is the i-th courier, M is the set of all the merchants, KS is the measurement of the Kolmogorov–Smirnov test, S_i^m is the set of reporting errors of c_i at the merchant m. Given *cscore*, we explore the K-means algorithm with a different number of clusters. The optimal number of the clusters is selected empirically based on the model performance on clusters, which finally results in 10 clusters.

4.1.3 ***Model Formulation for Arrival Time Prediction.*** We explore several machine learning algorithms and finally select a multi-layer neural network [42] to predict arrival time for each cluster for two reasons: (i) high accuracy, and (ii) easy to update with new coming data. Especially, the updating ability is necessary for our following modeling process in inside-merchant reporting. Formally, we present the training process as finding

$$w^*_{base} = \arg\min_w \frac{1}{N} \sum L(\phi(x_i, w), y_i) \quad (2)$$

where w is the learning parameter, N is the number training samples, L is the loss function such as the mean squared error, x_i and y_i is the feature and label of the i-th sample, ϕ is the training model (i.e., a neural network), which is implemented with 8 hidden layers with 48 nodes in each layer in PyTorch [35].

4.1.4 ***Label Extraction for Model Training.*** In order to train the above model to predict actual arrival time, we need labels. *If*

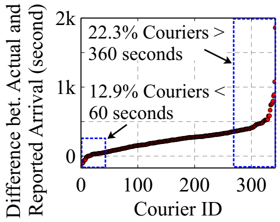


Fig 6: Average Reporting Error of Couriers

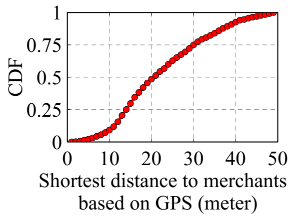


Fig 7: Distance between Couriers and Merchants

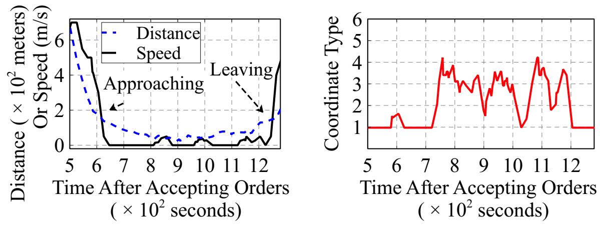
we assume the actual arrival time can be obtained accurately from the historical GPS trajectories, we can use historical arrival time as labels for model training. We first extract the arrival time from the trajectories. Ideally, if a courier has arrived at a merchant, the distance to the merchant is close to zero. However, in practice, the distance may vary because of the erroneous coordinates of the couriers or the merchants. In Fig. 7, we plot the shortest distance between couriers' and outside-merchants' coordinates and found that half of the shortest distances are more than 20 meters. To address the problem, we explore the idea that even the nearest distance is larger than zero, but the distance follows a common trend, i.e., changes from decreasing to increasing, where the shortest distance in the tread corresponds to the arrival time. In this way, we obtain a stable estimation of the arrival time without introducing any empirical distance threshold. However, the key assumption, i.e., the actual arrival time can be obtained accurately from the historical coordinates, is true for outside-merchants, but not true for inside-merchants due to the large error of GPS indoor, which will be addressed in our advanced model as follows.

4.2 Advanced Model for Arrival Time Prediction at Inside-Merchants

4.2.1 ***Why the base model is not enough?*** Given the base model introduced in the last subsection, a straightforward approach is to directly apply the learned model based on the same features extracted from the inside-merchant reporting. We tested this approach in the inside-merchant reporting scenario but approved it introduced much higher errors (details in § 7). The reason for the failure is because of the key difference between picking up from outside-merchants and inside-merchants that couriers still need to walk in the building for a while. If we directly use the base model, the predicted arrival time is closer to the building entering time than the merchant arrival time inside the building. This difference makes the model learned by outside-merchant labels do not work well in the inside-merchant scenarios.

4.2.2 ***New Constraint for Inside-merchant Scenarios.*** The difference between outside-merchants and inside-merchants also brings us an opportunity to address the problem, which introduces a practical constraint that couriers' arrival time *cannot* be earlier than the building-entering time. In this case, we could potentially use this constraint to adjust the predicted arrival time. To make use of this factor, we first detect the building entering time and location of couriers. More generally, we detect (1) the indoor/outdoor switching time, and (2) the indoor/outdoor switching location. In addition to time, we also need the switching location because couriers may enter the building from different entrances.

(1) Indoor/outdoor Switching Time: In the previous work, Li *et al.* proposed to utilize various smartphone sensors to detect if a user is in the indoor or outdoor [26]. Considering the design goal of transparent localization, we only utilize the existing localization module to detect the indoor/outdoor switching based on the data from couriers' trajectories.



(a) Distance and Speed **(b) Coordinate Type**

Fig 8: Couriers' signals when they are near to the merchant including three observed features: (a) distance to the merchant and speed; (b) coordinate type (in Table 2) from smartphone localization modules.

Fig. 8 plots the observations of the courier's status near to the pickup merchant. The distance line in Fig. 8a shows the distance between the courier and the pickup merchant, which indicates the period from approaching the merchant (i.e., distance decreases) to leaving the merchant (i.e., distance increases). Note that the distance is measured by the GPS and can have great offsets when near to the building. Similarly, the speed line has a similar trend. In addition, we observe the distinct signal changes between approaching and leaving the building in Fig. 8b that the value of the coordinate type increases to more than one and fluctuates. Based on these observations, we perform a two-step filtering process: (i) we extract the target period based on the distance and speed signal. Specifically, we detect the "plateau" in both distance and speed such as from 600 seconds to 1200 seconds in Fig. 8a, which gives a rough period when the switching happens. (ii) based on the "plateau", we use a sliding window approach (i.e., duration of one minute) to determine if a courier is in an outdoor or indoor environment. In each sliding window, we extract a few empirical features including:

- Minimum value of speed, coordinate type, distance;
- Maximum value of speed, coordinate type, distance;
- Average value of speed, coordinate type, distance;
- Standard deviation of speed, coordinate type, distance;
- First and last observation of the distance in the period;

We collect 7,000 data samples and manually label their indoor and outdoor status by comparing the trajectories in each sliding window to the building polygon. Based on the features and labels, we train a random forest model [4]. In five-fold cross-validation experiments, it shows 92% of the average accuracy. Based on the detected status, we search two consecutive windows with changing indoor/outdoor status (e.g., from the outdoor status to the indoor status) and average the window start time as our indoor/outdoor switching timestamp.

(2) Indoor/outdoor Switching Location: To find the indoor/outdoor switching location, the straightforward solution is to extract the GPS locations in the corresponding outdoor/indoor switching timestamps. However, under testing, we found the average distance between the switching location extracted by GPS and the gate of the building is 44.8 meters. The main reason is that (i) the GPS signal becomes volatile when near to the tall building; (ii) the GPS uploading frequency is 20 seconds, which counts for 28 meters offset if considering the average walking speed as 1.4 meters/second [5]. To reduce the distance error, we apply a DBSCAN algorithm to cluster these switching locations into several clusters. Compared with other clustering algorithms such as K-means, DBSCAN is more effective for spatial clustering. Without concerning the number of clusters, we only need to control the maximum distance between points in the same cluster. Based on the clustering result, we assign the clustering centers as our new switching locations, and the result shows the distance error is reduced to 19.2 meters. Because we shift the switching location, the corresponding switching time should also be adjusted by the walking time between the previous and new switching locations. Considering a short walking distance (e.g., a few meters) between switching locations, we assume the walking time is calculated by the distance between the switching locations divided by the average walking speed.

Note that the intuition of clustering is to align couriers to the same location before entering the building. In practice, instead of the actual gate, we could make this "same location" as any location near to the gate (e.g., a few seconds of walking distance) that couriers have the same distance to the gate.

4.2.3 ***Model Adaption.*** Given the new constraint of switching time and locations, it is not trivial to incorporate this constraint to the previous model ϕ because it does not exist in the outside-merchants. In fact, the switching time in the outside-merchants is numerically equivalent to the label, which would introduce overfitting if considering it as a feature. To address the problem, we design a model adaption approach based on transfer learning [33] with hard constraints. The key idea is that we keep the base model but explicitly incorporate the new constraint to the model tuning process with new samples (i.e., approximated actual arrival time introduced next paragraph) from the inside-merchant reporting scenario. Formally, we present the adaption process as finding

$$\begin{aligned} w^*_{advance} &= \arg\min_{w} \frac{1}{K} \sum L(\phi(x_i, w), y_i) \\ & s.t. \phi(x_i, w) > e_i \text{ for } 1 \leq i \leq K \end{aligned} \tag{3}$$

where K is the number of new samples from the inside-merchant reporting scenario, e_i is the entering time for x_i, $\phi(x_i, w) > e_i$ performs as the constraint.

(1) Approximated Labels for model training related to inside-merchants: We perform the same feature extraction process as the outside-merchant reporting behavior. However, compared with the outside-merchant reporting behavior, there is no explicit data to be considered as the training labels because of the unreliable GPS of couriers when they are indoor. To address this issue, our intuition is that when a courier enters and leaves a building in a relatively short time Δ (e.g., the merchant is near to the entrance of the building), the arrival time as labels can be approximated by the median time between her/his entering and leaving the building. We further refine these actual arrival times by collecting the ones with no waiting time since we know the courier departure from inside-merchants as well as shown in Fig.1. To verify our intuition, we first show the indoor duration between the switching time (i.e., from entering a building to leaving a building) in Fig. 9. We found

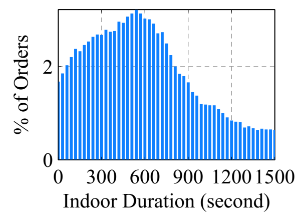


Fig 9: Indoor Duration

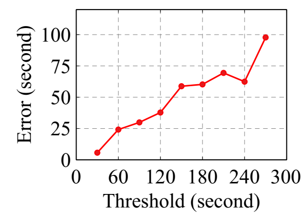


Fig 10: Threshold

there are 1.6% of the orders have a duration lower than 30 seconds (Δ) (i.e., the first bar in Fig. 9), which has the potential to be our approximated labels. We compare these approximated labels with separately-collected ground truth (introduced in Implementation) under different thresholds in Fig. 10. The x-axis is the threshold Δ to select approximated labels and the y-axis is the average difference between the approximated labels and the ground truth. We found the approximation error is 5.7 seconds given the threshold of 30 seconds, which provides feasible labels for our modeling.

(2) Model Training: In the training process, we perform a parameter-level transfer learning [33] that the initialized parameter w is the learned parameter w^*_{base} from the base model. Given the constraint, Eq. 3 cannot be easily solved with conventional optimization algorithms such as Stochastic Gradient Descent. To solve Eq. 3, we utilize a conditional gradients method [38] and the details are omitted given limited space.

5 COURIER LOCALIZATION FOR INSIDE-MERCHANTS

5.1 Intuition on Indoor Walking time

To implement predictive indoor localization, when a courier heads to pick up orders, we ask two questions: (i) what merchants will the courier pass? (ii) When will the courier pass each merchant?

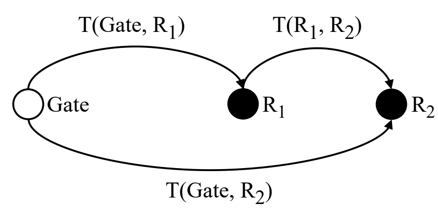


Fig 11: An example with three historical orders.

Take an example in Fig. 11 with one building gate and two merchants R_1 and R_2. We denote $T(\cdot,\cdot)$ as the duration between two arriving timestamps. $T(Gate, R_1)$ and $T(Gate, R_2)$ represent the walking time to R_1 and R_2 from the building gate. When a courier picks up orders in R_2, we are interested in if the courier passed R_1. If so, we predict that a courier would be near to R_1 at time $T(Gate, R_1)$ on the way to R_2. Given the intuition, we solve two problems including the walking time and the visiting order.

5.2 Indoor Walking Time Estimation

In our setting, we have two kinds of walking time: (i) time from entrances (e.g., building gate) to merchants, and (ii) time from merchants to merchants. Based on the switching time and predicted arrival time, we could directly obtain the walking time from entrances to merchants. However, we found, due to the limited visited merchants of each courier, the obtained walking time is not complete. Fig. 12 plots the percentage of merchants that a courier visited in a building. We found most couriers only visited less than 10% of the merchants. Another peak appears in 60% for those couriers working frequently in this area. If we create a two-dimensional matrix with the size of *#couriers* × *#merchants* to represent the walking time of each courier to each merchant, then the matrix sparsity would be 77% in our scenario. In addition, the walking time from merchants to merchants is also not complete. Fig. 13 shows the pickup merchants where a courier visited after entering a building. We found a courier would pick up orders from multiple merchants with a probability of 32%, which leads to movements between merchants. For the rest of 68%, we cannot obtain the walking time from merchants to merchants.

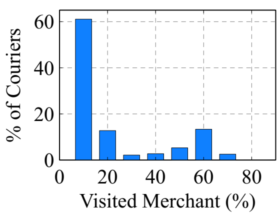


Fig 12: Visited Merchant

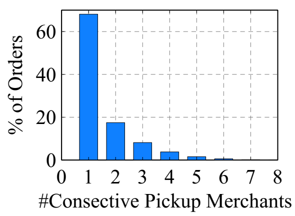


Fig 13: Multiple Visiting

Motivated by the data sparsity problem in recommendation systems, we infer the missing walking time in a collaborative filtering manner. Suppose we have C couriers and M merchants. Based on these observations, we construct two kinds of walking time matrices: a courier-merchant matrix X with C rows and M columns where each entry is a courier's walking time from the gate to a merchant; a merchant-merchant matrix Y with M rows and M columns representing the walking time between all M merchants for a particular courier (we have C of these matrices Y). In a general approach, we apply non-negative matrix factorization (NMF) to each of the matrices with the multiplicative update (shown as Eq. 4).

$$minimize||X - WH||_F^2, minimize||Y - RS||_F^2, s.t. W, H, R, S \geq 0$$
$$W = W \cdot \frac{XH^T}{WHH^T}, H = H \cdot \frac{W^T X}{W^T WH}, R = R \cdot \frac{YS^T}{RSS^T}, S = S \cdot \frac{R^T Y}{R^T RS} \quad (4)$$

Based on W, H (or R, S), we train a neural network [42] to predict missing walking time. The training input is the concatenation of W and H (or R and S) in the corresponding cell, and the training labels are the known walking time from predicted arrival time (or visiting between merchants). Our implementation is based on a neural network with 8 hidden layers and 32 nodes in each layer, which is trained in five-fold cross-validation experiments. Specifically, we randomly split the known walking time into five-fold and choose one-fold as a testing set when training the neural network.

5.3 Symbolic Mobility Graph Construction

Based on the complete walking time, we design a solution to infer the visiting order based on a symbolic mobility graph, where each node is a merchant and each edge represents a path between them. The weight of each edge is the walking time. The advantage of a symbolic mobility graph lies in its simplicity without introducing complex environmental factors such as routes and elevators. This is motivated by the work done by Wang et al. [51] where they try to estimate vehicular travel time without specifically discussing factors such as routes and traffic lights. We encapsulate indoor factors into the walking time between nodes instead of explicitly including them in the design.

Algorithm 1: Symbolic Graph Construction

Input: Matrix X, Matrix Y, threshold α
Output: Mobility Graph

T_x, T_y : walking time matrices;
$T_p(p)$: walking time on path p; P as an empty path set
Sort all the nodes by T_x and store in a min-heap H.
A = $H.pop()$, $P.add(Path(Entrance, A))$
while H *is not empty* **do**
  next = $H.pop()$
  $min\_p = argmin_{p \in P} |T\_x(next) - T_p(p) - T_y(p.tail, next)|$
  $min\_err = |T\_x(next) - T_p(p) - T_y(p.tail, next)|$
  if $min\_err < \alpha$ **then**
    new_path = $min\_p + next$
    $P.add(new\_path)$
  else
    $P.add(Path(Entrance, next))$

Our idea is that if a merchant appears on the path to another pickup merchant, then total walking time should be equal to the sum of the two sub-period walking time. Take an example in Fig. 11. If R_1 is on the path to R_2, then $T(R_1, R_2) \approx T(Gate, R_2) - T(Gate, R_1)$. To this end, we design an iterative approach to find the path to all the merchants. We first sort all the nodes and select the one with minimum walking time in X as our initial node. It is straightforward that there is no previous node before the initial node since it is the minimum walking time. Then we set our first path from the entrance to the initial node. For the following nodes, we iteratively check existing paths and choose the one satisfying our approximation error of α. Otherwise, we consider there is a direct path from the entrance to the node without passing other nodes. The detailed algorithm is shown in Algorithm 1. Different from the multi-dimensional scaling method [56] that only provides relative locations, we also infer the visiting order between them.

5.4 Real-Time Predictive Localization

We introduce our algorithm for real-time localization. The input includes the pickup merchant (obtained by current orders), real-time GPS, walking time matrix of the courier, and the symbolic mobility graph. As the courier moves near to the building (i.e., 500 meters), we start to detect when the courier enters the building to determine the switching location O and time t_o. Then we retrieve the corresponding walking time matrix and the symbolic mobility graph. Based on the path from the symbolic mobility graph, we determine which merchant is the closest one in real time to provide the merchant-level localization.

6 IMPLEMENTATION

(1) System Overview. We implement a prototype system in the [anonymous] platform, one of the largest food delivery platforms

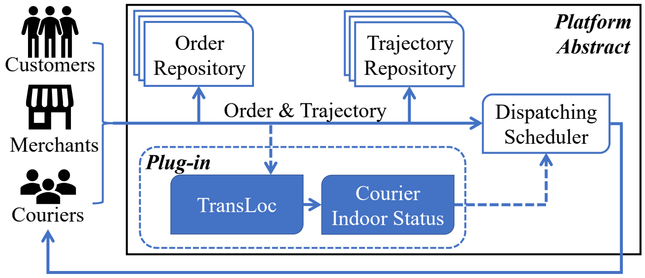


Fig 14: Instant Delivery Platform with TransLoc

in Chinese city Shanghai. Fig. 14 plots the abstract but general components in instant delivery platforms, where customers, merchants, and couriers upload the real-time order status and trajectories to the platform. Based on this information, the platform dispatches incoming orders to different couriers every 30 seconds. Without changing the existing platform structure and workflow, our system *TransLoc* performs as a plug-in component to predict the couriers' indoor locations based on the same real-time information, which is then sent to the dispatching scheduler component to optimize the dispatching strategy.

(2) Ground Truth of Indoor Locations/Delivery Time. As a prototype, we select a shopping mall with 65 merchants as the testbed, among which we obtain the consents from 27 merchants to deploy Bluetooth beacon devices. The devices are based on model nRF51822 working with a frequency of 800ms and transmission power of -60dBm. Each Beacon device is registered with a unique id and is bind to one and only one merchant, and continuously broadcast Bluetooth signals. We deploy them in merchant entrances as there are minimal obstacles between couriers and beacons to reduce environmental effects. The courier's smartphone detects the Beacon device id (i.e., UUID) and the signal strength, which implies the courier is near to the merchant with the beacon. Considering the smartphone heterogeneity and possible unstable signals, we follow the idea in [40] that it is more stable to localize by the signal strength trend, i.e., changes from increasing to decreasing, instead of a threshold with a concrete value. In this way, we collect the ground truth as the time when the detected signal strength reaches to the lowest in the trend. Note that this ground truth serves for evaluation of both the indoor locations and delivery time.

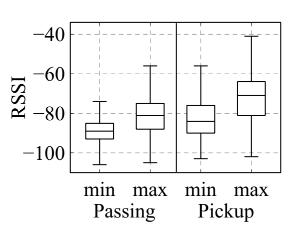


Fig 15: Min/Max RSSI at Merchants

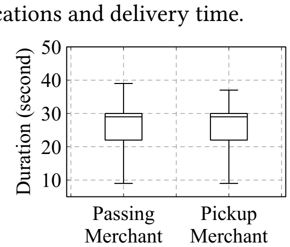


Fig 16: Detected Duration at each Beacon/Merchant

(3) Ground Truth Analysis on Merchants and Waiting Time. We analyze collected ground truth by the received RSSI from the beacons in the passing merchants and pickup merchants. Fig. 15 shows the min and max detected RSSI from beacons in passing and pick merchants. The observation is that the min/max value of RSSI in the pickup merchants is higher than that in the passing merchants. This is because the couriers generally pick up food inside the merchants, which makes them closer to beacons. In addition, we analyze how long a beacon is continuously detected both in passing merchants and pickup merchants in Fig. 16. We found the duration distribution in the passing merchant and pickup merchant is very similar, which indicates most couriers do not need to wait at the merchants when picking up and they move similar to the passing merchants. It also implies that there is a limited impact of the waiting time in indoor walking time.

7 EVALUATION

7.1 Methodology

7.1.1 ***Setting.*** We train the reporting model based on the reporting behaviors from outside-merchants and a few estimated training labels from inside-merchants, both of which are completely different from the ground truth collected from beacon devices in our test; (ii) in our training process, we follow the standard of dividing the dataset into training, validation, and test set to avoid overfitting.

7.1.2 ***Metrics.*** We first evaluate the reporting behavior modeling. The metric is to compare the predicted arrival time with the ground truth and defined as the absolute error in Equation 5,

$$AE_i = |y_i - \hat{y}_i| \tag{5}$$

where $\hat{y}_i$ and y_i is predicted arrival time and the ground truth of the i_{th} order, respectively.

For indoor localization, we are interested in two metrics that matter the most for the platform: (i) detection rates: how many passing merchants are correctly detected? (ii) for those detected merchants, what is the error between predicted time and ground truth? To this end, we define two metrics: the detection rate (DRate) in Equation 6, and the absolute error (AE) between the predicted time and the ground truth (the same as Equation 5).

$$Detection\ Rate\ (DRate_i) = \frac{\#(Detected\ Merchant)_i}{\#(Passing\ Merchant)_i} \tag{6}$$

7.1.3 ***Baselines.*** We introduce three baselines for the evaluation, each of which is the representative work of localization in different categories. We do not include the methods required sophisticated devices such as ultra-bandwidth [29], or antennae arrays [54], considering its practical constraint in large-scale deployment.

- GPS based method(GPS): We utilize received GPS signals to localize couriers, which is considered as the low-bound performance in indoor localization. If the GPS coordinate is within 10 meters of the merchant location, we consider it as correct detection. The AE is calculated by the ground truth to the time when being closest to the merchant.
- Wi-Fi Crowdsourcing(WFC) [63]: A widely used commercial method for indoor localization is based on Wi-Fi crowdsourcing. Companies maintain a database mapping the Wi-Fi access points to observed GPS coordinates from crowdsourcing. When a device scanned an access point existing in the database, we retrieve its GPS coordinates. In our work, shown in Table 2, coordinates with Type 5 are exactly obtained by Wi-Fi crowdsourcing, which are served as the localization result based on WFC. Since WFC localization also returns GPS coordinates, we use the same DRate and AE calculation method as the GPS baseline.

- Fingerprinting/landmark-based method(FP) [50]: This method is widely studied in academia but is deployed on a large scale because of the expensive cost of collecting fingerprinting databases in different buildings. In our scenario, we randomly select beacons as pre-collected fingerprinting locations. That is, when couriers are near to these fingerprinting locations, we can know their locations. Other locations between fingerprinting locations are estimated by dead-reckoning on average walking speed [50] (e.g., we assume the indoor map is available in FP baseline). Introducing inertial sensors could definitely improve the performance. However, the purpose of our fingerprinting based baseline is to show if our system without additional labor can perform similarly compared with the labor-intensive method. From this perspective, we do not introduce more delicate baselines in the paper.

Note that the baseline FP cannot be well fit our actual business scenario because of the large-scale deployment constraint. For the purpose of comparison, we assume the availability of required information in the baselines. The comparison may be unfair to our system but could be considered as a measurement that how close our method is approaching the performance of the baselines with expensive deployment. Collecting more smartphone sensor data could improve performance. However, our approach can serve as a lower-bound solution without any lower-level sensor data input to explore what is the performance in this setting.

7.2 Overall Performance

We show the performance of reporting behavior modeling in Fig. 17 including both inside-merchants and outside-merchants. The coordinate zero in X-axis represents the range between 0 to 10 seconds. We found the absolute error of the system on outside-merchants is 112 seconds, and that on inside-merchants is 133 seconds. When we dig into the error distribution of inside-merchants, it shows 42% predictions have errors lower than 10 seconds, and 60% is lower than 30 seconds. The large mean error is mainly because of the errors in the skewed long-tail. To show the effect, we use our predicted arrival time to re-compute the reporting errors in Fig. 18, compared with the original reporting errors. The result shows the percentage of reporting errors within 1 minute is increased by 57%.

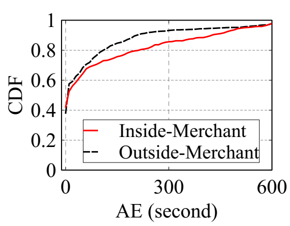


Fig 17: Absolute Error on Reported Arrival

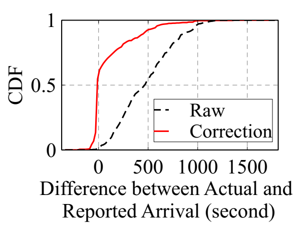


Fig 18: Reduced Error on Reporting Error

We then evaluate the detection rate *DRate*. Fig. 19 plots the CDF of the detection rate of TransLoc compared with baselines with the order is FP > TransLoc > GPS > WFC. More specifically, the average detection rate of TransLoc is 0.45, compared with 0.49 of FP, 0.06 of WFC, and 0.31 of GPS. We found TransLoc's performance is close to FP's, which proves its effectiveness even without the expensive deployment cost of FP. Surprisingly, we found the detection rate of GPS is higher than WFC. This is because the WFC has only limited recorded access points in the database compared with easily accessible GPS, which limits its detection ability.

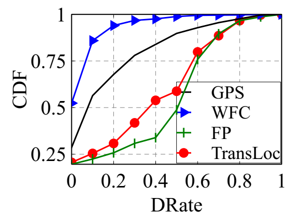


Fig 19: Detection Rate

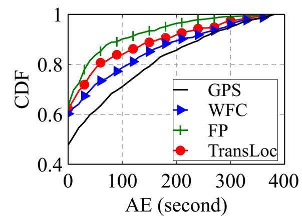


Fig 20: Localization Error

We further evaluate the localization error. Fig. 20 plots the errors of our system and baselines. The performance order is FP > TransLoc > WFC > GPS. Among the localization error, if we consider AE < 30 seconds as correctness, we found 24% of the results are overestimated and 19% are underestimated, which are evenly distributed around the ground truth. Quantitatively, the mean error is 31 seconds of FP, 42 seconds of TransLoc, 126 seconds of WFC, and 162 seconds of GPS. It shows TransLoc improves the WFC and GPS methods by 64% and 72%, which has close performance as FP. Similarly, we found TransLoc's accuracy is close to FP's. Compared with those two methods, WFC and GPS based methods have higher errors. This is because the WiFi signal has a high penetration ability, which leads to early detection. GPS's low precision is mainly because of its natural weakness in indoor environments.

7.3 Importance of Technical Components

We analyze the importance of our technical components by comparing the results between our model and several variations including:

- *TransLoc* without transfer learning (**w/o-transfer**): A key technical contribution of our system is to use transfer learning to model human behavior in different scenarios. In this variation, without transfer learning, we directly use the behavior model learned from the outside-merchant scenario without adaption.
- *TransLoc* without the new factor constraint (**w/o-constraint**): The entering time and location is an important factor to optimize the model. In this variation, we implement the transfer learning based model adaption without the introduced constraint.
- *TransLoc* without grouping (**w/o-grouping**): We show the effect of grouping by a variation without grouping.

We show the comparison in Fig. 21 and found $wt - transfer$ is the variation with worst performance, which implies the transfer learning component is of great importance. The lesson we learned is that even most couriers are statistically consistent in different scenarios (shown in Fig. 4b), it cannot lead to good experiment performance without careful adaption. Further, both the constraint and grouping component result in a positive impact on the system.

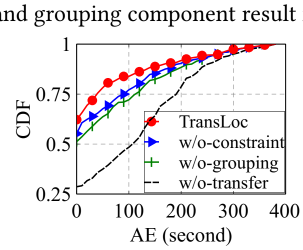


Fig 21: Variations

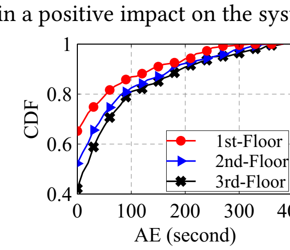


Fig 22: Different Floors

7.4 Evaluation on Robustness

We evaluate the robustness of our system from two perspectives: the environmental impact and the human impact. Specifically, we consider (i) the pickup merchants on different floors; (ii) couriers with different experiences.

Performance on different floors: Fig. 22 shows the localization performance when couriers head to pickup merchants on different floors. We found our system performs the best when on the first floor because of the less negative environmental impact such as elevators. In Fig. 22, the main difference exists in the low error areas, i.e., 0-100 seconds, that our system is more likely to achieve low errors on the first floor than others.

Performance on couriers with different experiences: Different couriers may have different behaviors if they are new to the environment, i.e., a shopping mall. To show the impact, we select couriers from two groups: (i) coming to the mall building only once in our observed week (new couriers); (ii) coming to the mall building every day (experienced couriers). Fig. 23 shows the comparison of those two groups of couriers that the system shows better performance on experienced couriers. The main reason is that (i) experienced couriers are familiar to the indoor environment better and have more stable behaviors; (ii) there is less data for the new couriers to learn the walking time matrix.

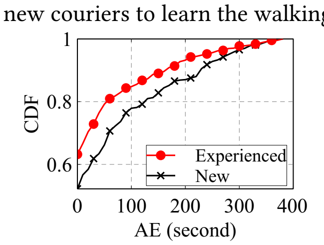


Fig 23: Experience Impact

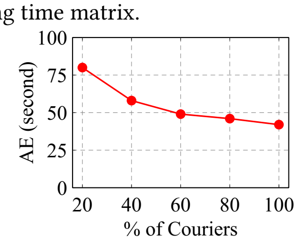


Fig 24: Scalability

7.5 Evaluation on Scalability and Overhead

We show the scalability of the system by showing its average performance in five-fold experiments with different percentages of couriers in Fig. 24. Compared with the full couriers, we found the system has a higher error when the percentage of couriers is low (e.g., 20%). The main reason is that it requires sufficient couriers to construct the walking time matrix using NMF. In a higher percentage of couriers (i.e., after 60%), the performance is more stabilized.

We further evaluate the overhead of our system. We implement our work in a server equipped with a Xeon CPU E5-1660 (only a single core is used) and a Tesla K40c. Thanks to the efficiently designed features, our model finishes training in our tested shopping mall with 20 seconds, which makes our system could be updated frequently with new data. In the real-time inference, the result is based on the path in the symbolic mobility graph and walking times, which makes it a linear time algorithm proportional to the number of nodes in the graph. More importantly, our system can be implemented mall by mall independently, which promises its ability for distributed deployment on a large-scale.

7.6 Application: Improving Platform Order Dispatching Performance

To evaluate the benefit of our system, we show how *TransLoc* can help optimize an order dispatch strategy with indoor localization. Our platform currently improves efficiency by assigning the orders from the same merchant to the same courier. For example, if the gaps between two orders from the same merchant are not significantly long (e.g., 10 minutes), these two orders would be dispatched to the same courier. Without indoor localization, a practical problem is that couriers may have left the merchant when receiving the second order from the platform, which forces couriers to go back to the merchant again and leads the redundant walking time. We analyze it by introducing a measurement *redundant time* (RT) calculated as *RT = Second Order Dispatching Time - First Order Pickup Time*, where *First Order Pickup Time* denotes the time a courier picks up an order from a merchant (e.g., m) for the first time after entering the mall; *Second Order Dispatching Time* denotes the time that the platform dispatches a second order of m to the same courier. The negative RT means both orders are dispatched before couriers' arrival, which does not introduce any redundant walking time. Otherwise, the positive RT means the second order is dispatched after couriers' first arrival, which indicates a go-back pickup with redundant walking time. We plot the CDF of RT in Fig. 25 (i.e., Raw RT) and found that most of the second orders are accepted later than couriers' first arrival time with 7 minutes on average.

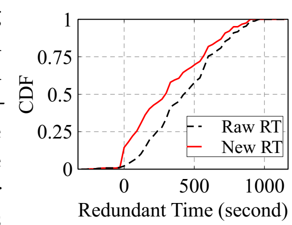


Fig 25: Redundant Time

To reduce the redundant walking time, we utilize the outputs from our system. As a case study, we add a constraint to the current dispatching strategy. When dispatching the second order, if the walking time between a courier's current indoor location and the pickup merchant is larger than a time threshold (i.e., 3 minutes in our study), we simply dispatch this order to other couriers who are heading to the merchant. Otherwise, if the walking time is less than 3 minutes or no other couriers are available, we did not change the dispatching that still makes the courier go back to pick up the order. Based on this strategy, we plot the experimental result as *New RT* in Fig. 25. We finally improve 21.3% of the orders and result in 349 seconds RT on average, which improves the raw RT by 24% and reduces the overdue rate by 2.5%.

8 RELATED WORK

Instant Delivery: Many works on the study of instant delivery problem focus on the products delivery [17] [31] [9], especially the recently increasing trend on the food delivery study [61] [23] [28] [58]. Ji et al. [23] provide a task grouping method for food delivery services to reduce the waiting time of users. Liu et al. [28] present a crowd-sourcing based food delivery approach that utilizes the taxis to support on-demand take-out food delivery. Yuan et al. [58] deploy a carpooling system for food delivery by a negotiation algorithm to improve the efficiency and users' utility. Compared with previous works focusing on scheduling optimization, our work utilizes an indoor localization method to improve delivery efficiency.

Indoor Localization: Many works on indoor localization utilize the static infrastructures, such as WiFi [49][30][52] [45] [53], Beacons [7][8][3], RFID [12] [29] [14], visible lights [59] [60] [44], etc. For example, Xie et al. [53] designed a device-free Wi-Fi tracking system. Zhang et al. [60] utilized ubiquitous visible lights for indoor localization. Ma et al. [29] implemented ultra-wideband localization for deployed RFID tags. The basic triangulating based localization approaches are not stable because of the multiple path problem [24] [49]. Even approaches are designed to address the problem, it is not clear how to apply on a large scale. Other approaches require pre-collected fingerprinting datasets or pre-deployed devices, which are expensive and not practical on a large scale. In our work, we did not rely on any extra deployment and is easy to be extended to a large scale. In addition, there are also many works based on the smartphones [36] [56] [25] [46] [62] [50] [37]. In these approaches, they generally fit in a few buildings with prior knowledge such as floor maps and not scalable to different environments. Further, continuous tracking requires a significant cost of energy [57], which generally cannot support couriers all day long. In our work, couriers are not required to perform any extra action so we do not introduce extra energy consumption. The most similar work is done by Yang et al. [55] focusing on localization on highways that is significantly different from our indoor environment.

9 DISCUSSION

Lessons learned: (1) The key lesson is that we can reduce infrastructure deployment by incorporating human behavior analysis and modeling. By comparing to the baseline of FP (i.e., in Fig. 20), we found our predictive approach can achieve a relative approaching performance. This insight gives an implication that we could use prediction methods to replace physical deployment in certain situations to reduce the overhead. (2) Based on our ground truth, we found that our system can achieve 10-second errors of arrival time prediction for more than 42% of orders as shown in Fig. 17, which reduces the reporting errors by 57% as shown in Fig. 18. It indicates that we can incorporate human behavior to design systems when it is highly predictable. (3) For the system robustness, we also found that both lower-level floors (Fig. 22) and more courier experiences (Fig. 23) have positive impacts on localization results, which guides our future system deployment. Our system also benefits from more couriers to reduce errors, and it suggests our system is potentially scalable to more couriers (Fig. 24). (4) Accuracy localization has the potential to improve the order dispatching by finding the most appropriate couriers as shown in Fig 25. It leads to the shorter delivery time and a lower overdue rate, thus higher efficiency and profit of the delivery platform, which is a strong motivation for the delivery platform to invest in couriers' indoor localization. (5) A major advantage of our system is that it does not introduce an extra burden to the main business. Our method only utilizes currently collected data from couriers' work smartphones with no effect on their regular work. We utilize existing techniques because of robustness and reliability, which also benefits easy maintenance.
Limitation: (i) In our studied period, the result is based on the general cases without specifically modeling the outlier cases, which works well in majority cases. As for outliers, we envision our system could adapt it by specifically modeling couriers' behaviors based on the data from an outlier context such as in the rush hour or a holiday. With that, we could apply the same system but with different models to implement localization. (ii)Our data is merchant/courier specific. As a business process for a courier or a merchant to join the platform, it is mandatory to collect the necessary data to enable the delivery service. Further, our study is based on a one-week period. For new couriers/merchants after one-week data accumulation, we envision our system could have the same performance on these couriers.
A/B testing: In our platform, our work is the first pilot study to provide indoor localization in a real setting. Instead of comparing different groups of couriers, our A/B testing is conducted by showing the scheduling efficiency with and without our localization output in the field study, which shows the improvement over the current scheduling service (thus, the benefit of our system).
Generalizability: We implement and evaluate our system in one real-world platform for one week with 565 couriers, 128 merchants, and 14,743 orders in one city. But we believe the key idea can be generalized and adopted in many commercial platforms under two conditions: (1) the users report their status under a context; (2) user's mobility is rather logically linear, i.e., couriers visit merchants sequentially one by one without other stops in the middle and can be linearly represented by their arrival time. For example, there are many instant delivery platforms such as DoorDash, InstaCart, and Amazon Primer Now, which almost share the same workflow as our tested platform. In other non-instant delivery scenarios, we could also find similar observations. For example, Uber drivers report they pick up customers in the urban canyon, which has low GPS accuracy [13], so need to be localized for order dispatching.
Ethics and Privacy: The order and smartphone data of couriers used have been anonymized and were collected by our platform, which is under the consent agreement of the couriers. In the agreement, the couriers are notified that their data will be collected and used for analyses to improve delivery efficiency. We obtain their consent because the couriers work for our platform. We do not use the GPS information in our dataset to track the detailed trace of the couriers, but only infer the arrival time of the couriers on merchants, which also benefits all couriers and customers. Further, under merchant owners' agreements, beacon devices only broadcast Bluetooth signals and did not reveal any personal information.
Data Release for Reproducibility: To encourage research in this direction, we will release the data to the community.

10 CONCLUSION

In this work, we perform the first study on couriers' reporting behaviors and indoor mobility for instant delivery services. We found most of their indoor/outdoor reporting behaviors are consistent, which motivates us to use machine learning models to correct their indoor reporting behaviors by the outdoor reporting model. Further, we design a symbolic mobility graph, which captures their indoor mobility without prior knowledge of the indoor environment. Based on the experiment results, our system can improve the localization accuracy by 64% and 72% compared with the Wi-Fi/GPS-based method and competitive accuracy with the fingerprinting-based method. With a case study, our system helps the delivery platform reduce the courier redundant time by 24%.

ACKNOWLEDGMENTS

This work is partially supported by NSF 1849238, 1932223, 1952096, and 2003874. We thank all the reviewers for insightful feedback to improve this paper.

REFERENCES

[1] Amazon. 2020. Amzon Prime Now. Webpage. (2020).

[2] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep Learning based Wireless Localization for Indoor Navigation. In *The 26th Annual International Conference on Mobile Computing and Networking*.

[3] Behnam Badihi, Jianyu Zhao, Siyan Zhuang, Olli Seppänen, and Riku Jäntti. 2019. Intelligent Construction Site: On Low Cost Automated Indoor Localization Using Bluetooth Low Energy Beacons. In *2019 IEEE Conference on Wireless Sensors (ICWiSe)*. IEEE, 29–35.

[4] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[5] Raymond C Browning, Emily A Baker, Jessica A Herron, and Rodger Kram. 2006. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of applied physiology* 100, 2 (2006), 390–398.

[6] Eyuphan Bulut and Boleslaw K Szymanski. 2013. WiFi access point deployment for efficient mobile data offloading. *ACM SIGMOBILE Mobile Computing and Communications Review* 17, 1 (2013), 71–78.

[7] Sudarshan S Chawathe. 2008. Beacon placement for indoor localization using bluetooth. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 980–985.

[8] Sudarshan S Chawathe. 2009. Low-latency indoor localization using bluetooth beacons. In *2009 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 1–7.

[9] Chao Chen, Daqing Zhang, Xiaojuan Ma, Bin Guo, Leye Wang, Yasha Wang, and Edwin Sha. 2016. Crowddeliver: planning city-wide package delivery paths leveraging the crowd of taxis. *IEEE Transactions on Intelligent Transportation Systems* 18, 6 (2016), 1478–1496.

[10] Mengjing Chen, Weiran Shen, Pingzhong Tang, and Song Zuo. 2018. Optimal vehicle dispatching for ride-sharing platforms via dynamic pricing. In *Companion Proceedings of the The Web Conference 2018*. 51–52.

[11] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. 2010. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 173–184.

[12] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. 2017. 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*. ACM, 222–234.

[13] Danny Iland, Andrew Irish, Upamanyu Madhow, and Brian Sandler. 2020. Rethinking GPS: Engineering Next-Gen Location at Uber. Webpage. (2020).

[14] Han Ding, Jinsong Han, Chen Qian, Fu Xiao, Ge Wang, Nan Yang, Wei Xi, and Jian Xiao. 2018. Trio: Utilizing tag interference for refined localization of passive RFID. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 828–836.

[15] DoorDash. 2020. DoorDash. Webpage. (2020).

[16] Eleme. 2019. Eleme. Webpage. (2019).

[17] Poorya Farahani, Martin Grunow, and H-O Günther. 2012. Integrated production and distribution planning for perishable food products. *Flexible services and manufacturing journal* 24, 1 (2012), 28–51.

[18] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 1459–1468.

[19] Min Gao, Xiao Zhang, Tao Zhang, Ci Chen, Zhouhong Wang, Zhihao Lu, Weijie Ding, and Jiang Ouyang. 2018. An Experimental Study on WeChat-Based Large Scale Indoor Localization System. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 330–338.

[20] Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. Predicting human mobility via variational attention. In *The World Wide Web Conference*. ACM, 2750–2756.

[21] Gaode. 2020. Gaode LBS Platform. Webpage. (2020).

[22] InstaCart. 2020. InstaCart. Webpage. (2020).

[23] Shenggong Ji, Yu Zheng, Zhaoyuan Wang, and Tianrui Li. 2019. Alleviating Users' Pain of Waiting: Effective Task Grouping for Online-to-Offline Food Delivery Services. In *The World Wide Web Conference*. ACM, 773–783.

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

[25] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. 483–494.

[26] Mo Li, Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, and Guobin Shen. 2015. IODetector: A generic service for indoor/outdoor detection. *ACM Transactions on Sensor Networks (TOSN)* 11, 2 (2015), 28.

[27] Qiulin Lin, Lei Dengt, Jingzhou Sun, and Minghua Chen. 2018. Optimal demand-aware ride-sharing routing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2699–2707.

[28] Yan Liu, Bin Guo, Chao Chen, He Du, Zhiwen Yu, Daqing Zhang, and Huadong Ma. 2018. FooDNet: Toward an Optimized Food Delivery Network Based on Spatial Crowdsourcing. *IEEE Transactions on Mobile Computing* 18, 6 (2018), 1288–1301.

[29] Yunfei Ma, Nicholas Selby, and Fadel Adib. 2017. Minding the billions: Ultra-wideband localization for deployed rfid tags. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 248–260.

[30] Alex T Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. 2014. Sail: Single access point-based indoor localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 315–328.

[31] Renaud Masson, Anna Trentini, Fabien Lehuédé, Nicolas Malhéné, Olivier Péton, and Houda Tlahig. 2017. Optimization of a city logistics transportation system with mixed passengers and goods. *EURO Journal on Transportation and Logistics* 6, 1 (2017), 81–109.

[32] Meituan. 2020. Meituan. Webpage. (2020).

[33] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[34] Postmates. 2020. Postmates. Webpage. (2020).

[35] Pytorch. 2019. Pytorch. Webpage. (2019).

[36] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 293–304.

[37] Niranjini Rajagopal, Patrick Lazik, Nuno Pereira, Sindhura Chayapathy, Bruno Sinopoli, and Anthony Rowe. 2018. Enhancing indoor smartphone location acquisition using floor plans. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 278–289.

[38] Sathya N Ravi, Tuan Dinh, Vishnu Suresh Lokhande, and Vikas Singh. 2019. Explicitly imposing constraints in deep networks via conditional gradients gives improved generalization and faster convergence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4772–4779.

[39] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. 2008. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 313–325.

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

[41] Shilpa Mete. 2020. Coronavirus Scare Spikes Online Orders, Amazon Sparks Delivery War. Webpage. (2020).

[42] Donald F Specht. 1991. A general regression neural network. *IEEE transactions on neural networks* 2, 6 (1991), 568–576.

[43] Xiaocheng Tang, Zhiwei Qin, Fan Zhang, Zhaodong Wang, Zhe Xu, Yintai Ma, Hongtu Zhu, and Jieping Ye. 2019. 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*. 1780–1790.

[44] Zhao Tian, Yu-Lin Wei, Wei-Nin Chang, Xi Xiong, Changxi Zheng, Hsin-Mu Tsai, Kate Ching-Ju Lin, and Xia Zhou. 2018. Augmenting Indoor Inertial Tracking with Polarized Light. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 362–375.

[45] Xinyu Tong, Fengyuan Zhu, Yang Wan, Xiaohua Tian, and Xinbing Wang. 2019. Batch Localization Based on OFDMA Backscatter. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–25.

[46] Yu-Chih Tung and Kang G Shin. 2015. Echotag: Accurate infrastructure-free indoor location tagging with smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 525–536.

[47] Uber. 2020. Uber Eat. Webpage. (2020).

[48] Marlin W Ulmer and Barrett W Thomas. 2018. Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks* 72, 4 (2018), 475–505.

[49] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*. 165–178.

[50] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 197–210.

[51] Hongjian Wang, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. 2019. A simple baseline for travel time estimation using large-scale trip data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 19.

[52] Chenshu Wu, Zheng Yang, Zimu Zhou, Yunhao Liu, and Mingyan Liu. 2016. Mitigating large errors in WiFi-based indoor localization for smartphones. *IEEE Transactions on Vehicular Technology* 66, 7 (2016), 6246–6257.

[53] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. 2019. mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[54] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th {USENIX} Symposium on Networked*

Systems Design and Implementation ({NSDI} 13). 71–84.

[55] Yu Yang, Xiaoyang Xie, Zhihan Fang, Fang Zhang, Yang Wang, and Desheng Zhang. 2019. VeMo: Enabling Transparent Vehicular Mobility Modeling at Individual Levels with Full Penetration. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom'19)*.

[56] Zheng Yang, Chenshu Wu, and Yunhao Liu. 2012. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 269–280.

[57] Ali Yassin, Youssef Nasser, Mariette Awad, Ahmed Al-Dubai, Ran Liu, Chau Yuen, Ronald Raulefs, and Elias Aboutanios. 2016. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys & Tutorials* 19, 2 (2016), 1327–1346.

[58] Qiming Yuan, Rong Zhao, and Jin Zhang. 2019. FoodCarpool: A Negotiation-based Carpooling System for Take-out Food Delivery. In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 188–193.

[59] Chi Zhang and Xinyu Zhang. 2016. LiTell: robust indoor localization using unmodified light fixtures. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 230–242.

[60] Chi Zhang and Xinyu Zhang. 2017. Pulsar: Towards ubiquitous visible light localization. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 208–221.

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

[62] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, and Feng Zhao. 2017. Travi-navi: Self-deployable indoor navigation system. *IEEE/ACM transactions on networking* 25, 5 (2017), 2655–2669.

[63] Yuan Zhuang, Zainab Syed, You Li, and Naser El-Sheimy. 2015. Evaluation of two WiFi positioning systems based on autonomous crowdsourcing of handheld devices for indoor navigation. *IEEE Transactions on Mobile Computing* 15, 8 (2015), 1982–1995.