ALWAES: an Automatic Outdoor Location-Aware Correction System for Online Delivery Platforms

DONGZHE JIANG, Peking University, China
YI DING, Alibaba Group, China and University of Minnesota, United States
HAO ZHANG, Alibaba Group, China
YUNHUAI LIU*, Peking University, China
TIAN HE, Alibaba Group, China and University of Minnesota, United States
YU YANG, Lehigh University, United States
DESHENG ZHANG, Rutgers University, United States

For an online delivery platform, accurate physical locations of merchants are essential for delivery scheduling. It is challenging to maintain tens of thousands of merchant locations accurately because of potential errors introduced by merchants for profits (e.g., potential fraud). In practice, a platform periodically sends a dedicated crew to survey limited locations due to high workforce costs, leaving many potential location errors. In this paper, we design and implement ALWAES, a system that automatically identifies and corrects location errors based on fundamental tradeoffs of five measurement strategies from manual, physical, and virtual data collection infrastructures for online delivery platforms. ALWAES explores delivery data already collected by platform infrastructures to measure the travel time of couriers between merchants and verify all merchants' locations by cross-validation automatically. We explore tradeoffs between performance and cost of different measurement approaches. By comparing with the manually-collected ground truth, the experimental results show that ALWAES outperforms three other baselines by 32.2%, 41.8%, and 47.2%, respectively. More importantly, ALWAES saves 3,846 hours of the delivery time of 35,005 orders in a month and finds new erroneous locations that initially were not in the ground truth but are verified by our field study later, accounting for 3% of all merchants with erroneous locations.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing design and evaluation methods**.

Additional Key Words and Phrases: Online delivery, CrowdSourcing, Localization, Machine Learning

ACM Reference Format:
Dongzhe Jiang, Yi Ding, Hao Zhang, Yunhuai Liu, Tian He, Yu Yang, and Desheng Zhang. 2021. ALWAES: an Automatic Outdoor Location-Aware Correction System for Online Delivery Platforms. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 107 (September 2021), 24 pages. https://doi.org/10.1145/3478081

1 INTRODUCTION

Online instant delivery is an emerging business for Gig Economy [38] that online orders (e.g., food) are delivered by Gig workers (e.g., couriers) from merchants (e.g., restaurants) to customers within a short time (e.g., 30

*Corresponding author.

Authors' addresses: Dongzhe Jiang, Peking University, No.5 Yiheyuan Rd. Haidian District, Beijing, China, dongzhe.jiang@pku.edu.cn; Yi Ding, Alibaba Group, Shanghai, China, University of Minnesota, Minneapolis, United States, dingx447@umn.edu; Hao Zhang, Alibaba Group, Shanghai, China; Yunhuai Liu, Peking University, No.5 Yiheyuan Rd. Haidian District, Beijing, China, yunhuai.liu@pku.edu.cn; Tian He, Alibaba Group, Shanghai, China, University of Minnesota, Minneapolis, United States; Yu Yang, Lehigh University, Bethlehem, Pennsylvania, United States, yuyang@lehigh.edu; Desheng Zhang, Rutgers University, New Jersey, United States, desheng.zhang@cs.rutgers.edu.

2474-9567/2021/9-ART107
https://doi.org/10.1145/3478081

minutes). This business grows rapidly with the emergence of several online instant delivery platforms worldwide, e.g., Prime Now [4], UberEats [60], Instacart [37], and DoorDash [20] in the U.S.; Deliveroo [16] in the U.K.; Meituan [18], JD [58], and Eleme [68, 72] in China. On such a platform, merchants' physical locations are essential to assign orders to the most suitable couriers to finish the delivery before the promised time to avoid an overdue [18, 68, 72].

In practice, these merchant locations are usually registered by the merchants themselves and are not always correct. Some wrong location records may be by manual errors or merchant re-locations, while others are due to intentional manipulations. For example, a merchant registers its location in a large shopping mall, which typically implies a better service and a higher food quality [8] compared to a residential area [35]. As a result, some merchants may intentionally register a wrong location (e.g., in a mall) to attract more online consumers who do not have to go in person and only need a courier to pick up the order. We call such location registrations *location frauds*.

This kind of location frauds persists because most couriers are part-time and lack the incentives to correct them proactively. First, when couriers arrive at the merchant's wrong location for the first time and cannot find the merchant, they would call the merchant and find out the real place and then go there to pick up the order. Second, the merchants who are conducting location frauds may also actively call the couriers to go to the real location by claiming that they just moved as long as they know who the couriers are in the merchant app. Next time, these couriers may directly go to this real location when getting an order to pick up from the same merchant.

A naive solution for the platforms is to ask the couriers to report wrong locations, but this is not effective in practice because (1) average couriers lack incentives for reporting because their main job is delivering (e.g., 25 orders per day); (2) some malicious couriers could even collude with merchants. For example, there are more than 8 million registered couriers nationwide on the platform we are working with in China. Most of them are part-time couriers with little incentive to report errors. These location frauds significantly impact online delivery platforms because of resultant inappropriate courier scheduling, deviated courier's routes, and unnecessary overdue compensations to customers, as shown in Sec.2.2. Thus, the platforms need to identify these location frauds and then find their real locations.

The state-of-the-art technical solutions for location frauds can be roughly classified into two categories: the dedicated and crowdsourcing approaches. For the dedicated approaches, the platforms hire professional investigators to conduct field studies, collect new data [54], and verify the merchant locations one by one. But these dedicated approaches are labor-intensive and too expensive on a large scale. In contrast, for the crowdsourcing approaches, some already-collected location data from ordinary users of a system (e.g., navigation apps) are explored to identify potential location frauds (e.g., GOLD panning [3]) under some incentives from platforms; but gathering large-scale location data requires significant financial incentives[39].

In this paper, we explore a different technical approach that leverages the opportunities offered by the *order update data* collected on the online delivery platform. In an order delivery service, the order update data log the spatiotemporal records of four major events of the delivery (details in Sec.2) to notify customers of the real-time order status. These spatiotemporal records are obtained by three infrastructures, i.e., courier smartphones, physical check-in beacon devices, and virtual check-in beacon devices (embedded in merchant smartphones under their consent). We explore different tradeoffs of their cost and reliability detailed in Sec. 3. These measurements will reflect how long it takes for couriers to travel among merchants. These travel time can be used to validate if one or more merchants' locations may be erroneous, and if so, what are the real locations.

What makes these spatiotemporal records from order update data interesting are their strengths and weaknesses to detect location frauds.

- The key strength is that they are obtained automatically and cover our platform merchants without couriers' additional efforts.These records are mandatory for platform accounting; hence they can detect location frauds for "free". Note that we only use the data already collected from couriers for accounting under their consent, and do not use the platform customer data. Even though couriers' data have been used in recent work such as TransLoc[64], which corrects the couriers' indoor location by modeling uncertainty for an online delivery platform. However, these work assumed that all merchants' registered locations are correct so they can correct courier locations, which is not true in practice.

- The key weakness is the uncertainty of couriers' mobility behaviors. For example, couriers may travel between the same merchants on different routes while visiting other places during the process, which cannot be explicitly measured by the order status data or even known to the platform, given the delivery sequence of multiple orders at couriers' discretion.

To explore both their strengths and weaknesses, in this paper, we design **ALWAES**, an **A**utomatic **L**ocation-a**WArE** correction **S**ystem based on the spatiotemporal measurement results from online delivery platform infrastructures for outdoor merchants. We implement and evaluate ALWAES on a real-world online delivery platform to show its practical impacts. To summarize, our main contributions in this paper are as follows.

- To our knowledge, we conduct the first work to explore order status measured by the infrastructure of online delivery platforms to identify and correct outdoor merchant location frauds automatically. We demonstrate the feasibility of automatic location fraud identification and correction by designing the ALWAES system. ALWAES explores real-world order status data from our online delivery platform in Shanghai, including 10,821,351 orders delivered by 23,604 couriers for 2,897,080 customers during 85 days. *We will release one month of the data used, including BLE sensing, and manual report data, after we can reveal our identity to benefit the IMWUT community*[1]*. The data is prepared to be anonymous with panned date information to protect privacy.*
- We explore three kinds of infrastructures with different performance and cost tradeoffs for five spatiotemporal measurement strategies for order update data collection, including (1) couriers' smartphones only to report order status manually; (2) dedicated physical beacon devices to report order status automatically; (3) merchant smartphones as virtual beacon devices to report order status automatically. We process the order status collected from these three infrastructures as unified spatiotemporal measurement records for four order delivery events. We build a travel distance model based on machine learning techniques with couriers' correlated mobility features to address the uncertain mobility behaviors between these events. We design a graph-based multi-iteration algorithm to localize all merchant locations as if we did not know their locations. Finally, we compare the locations we infer and the locations registered by merchants to identify the potential location frauds.
- To explore the pros and cons of three infrastructures, we evaluate ALWAES with five different spatiotemporal measurement strategies for order status data from 23,604 couriers' phones, 3,200 physical check-in devices, and 2,790 virtual check-in devices in Shanghai. Compared to three baselines, the experimental results show that ALWAES improves the location fraud detection and corrections by 32.2%, 41.8%, and 47.2%, respectively, in terms of AUC given manually collected ground truth. For different measurement strategies within ALWAES, we found physical check-in devices combined with merchants' smartphones perform the best. More importantly, ALWAES has real-world impacts by saving 3,846 hours of the delivery time for 35,005 orders in a month and identifying eight location fraud merchants that are not in the ground truth but were verified by our field studies.
- We summarize some key insights on the different infrastructures (Details in Sec.6). (1) The accurate timing from physical devices outperforms all the other approaches in ALWAES but with an extra deployment cost. (2) The less-expensive virtual devices with only software development can achieve similar performance to

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=107267

physical devices in our system to save cost but need a high merchant participation rate. (3) The implicit and inaccurate couriers' manual input timing is helpful in ALWAES to avoid an extra hardware cost or participation dependence, with an inevitable reduction in accuracy compared to explicit timings from physical or virtual devices.

The rest of the paper are organized as follows. Sec. 2 presents the motivation for our work. Sec. 3 shows systems and data we used. Sec. 4 gives the details of ALWAES. Sec. 5 evaluates the performance, along with some discussions in Sec. 6. In Sec. 7 we review the related work. Sec. 8 concludes our work.

2 MOTIVATIONS

2.1 Online Delivery Platform

2.1.1 Platform Components. There are four basic components of an online delivery platform: a *customer* orders a *merchant*'s goods or meals online (e.g., smartphone apps or websites) through a *digital platform* (i.e., the online delivery service provider); a *courier* is assigned by the platform to pick up this order at the merchant and then deliver it to the customer. If the customer receives goods/meals later than a time limit, e.g., 30 minutes, an overdue occurs, and then the platform will compensate the customer for missing deadline. Couriers in dense urban areas usually prefer electric bikes (e-bikes) instead of cars to avoid traffic jams.

2.1.2 Order Update Data. During a delivery process, the platform will track the real-time status of the order, namely *order update data*, and log them for accounting and show them to customers for better customer experiences. An order update record logs four major events related to a courier from an order being placed until the order being delivered, (1) accepting the order, (2) arriving at the merchant, (3) departing from the merchant (with orders), and (4) delivering to the customer. We list the fields we use in this paper in Tab. 1.

Table 1. Order Update Data Format and Example

Field	Value
Order/Courier/Merchant ID	O001/C001/M001
Registered Merchant Location	116.418065, 39.916998
1. Acceptance Time & Loc	18-01-01 12:00:00 & GPS
2. Arrival Time & Loc	18-01-01 12:10:00 & GPS
3. Departure Time & Loc	18-01-01 12:10:10 & GPS
4. Delivery Time & Loc	18-01-01 12:25:00 & GPS

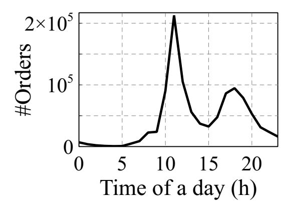


Fig 1. Orders During a Day

In our analysis, we found that a courier normally picks up multiple orders from different merchants altogether and then delivers them to customers one by one, given the large number of orders they deliver per day, especially during peak hours. Fig. 1 shows the number of orders during different slots of a day in Shanghai. Based on the departure time of one order and the arrival time of the next order, we obtain the travel time among merchants.

2.2 Locations Frauds and Their Impacts

2.2.1 Location Frauds and Why They Exist. In practice, the merchants locations are registered manually by the merchants themselves (e.g., adding a text address, uploading mobile devices' GPS location or picking a point on the map) and transformed into geographic coordinate by our system (e.g., the registered location by McDonald's (Wangfujing Street, Beijing) is "116.418065,

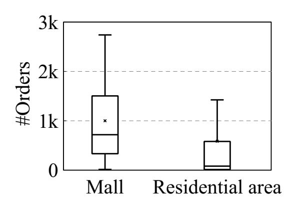


Fig. 3. Order Quantity in a Month

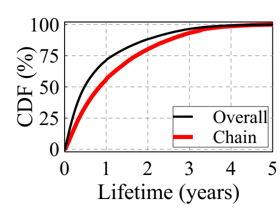
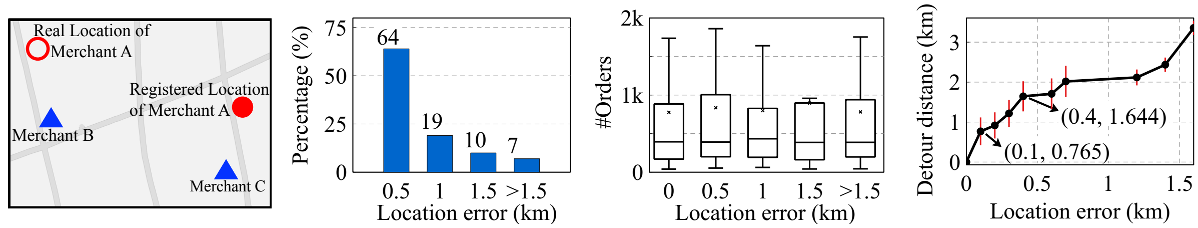


Fig. 4. Merchant Lifetime



(a) Location Fraud Example (b) Location Error Distribution (c) Merchants' Order Quantity (d) Average Detour Distance

Fig. 2. Impact of Location Frauds. (a) shows an example of location fraud. The red solid dot is the *registered location* of a merchant A; whereas the red hollow dot is its real location. When there is a new order for this Merchant A, if the platform knows there are two couriers in Merchant B and C, respectively, under this location fraud, the platform will mistakenly dispatch the courier at C but the courier at B is actually closer. (b) shows the distances between registered and real locations of the merchants we identify with potential location frauds based on our field studies. Over 36% of these merchants drift more than 500 meters; (c) shows the number of orders per month for merchants with different errors. We found that merchants with erroneous locations have a similar amount of orders as ones with correct locations (i.e., with 0km error), implying that they are equally important and treated in the online delivery system; To quantify their impact, (d) shows the relation of the merchant location errors against the courier detour distance of individual orders caused by erroneous locations.

39.916998"). As a result, the merchants' locations are easy to falsify manually. Fig. 2(a) gives a real example of merchants with a severe location error.

The goal of fraud merchants is to register a wrong location for attracting more online consumers who do not have to go in person and only need a courier to pick up the order. For example, a merchant registers its location in a large shopping mall which typically implies a better service and a higher food quality [8] compared to a residential area [35]. As a result, shops in a mall will have more orders, as shown in Fig. 3 which compares the order quantities between shops in the shopping mall and in the residential area. Intuitively, the merchants should care about long-term reputation to avoid any frauds and resultant consequences. However, we found in our platform, the lifetime of merchants is rather short due to the convenience and low-cost of registering an online shop in a platform. The CDF of merchant lifetime on our platform is in Fig. 4. Around 74% of merchants left the platform within 1 year but the chain stores typically have a longer lifetime. This is a rather interesting phenomenon for online shop lifetime, but a detailed study is out of scope of this work and worth its own study. Further, a platform rarely fines or bans the merchants with potential location frauds in practice because it is hard to prove the fraud and it hurts the platform's market share. We believe these are two core reasons for location fraud.

2.2.2 Attack Model. In our attack model, we consider three versions that the merchants registers the wrong location. Adversarial merchants want to attract more online orders through location frauds. They may get 665 orders on average per month during location fraud and only 521 orders on average in real locations based on the field studies (details in Sec. 3.3). (i) *Individual attack.* An adversarial merchant conducts location frauds on his/her own. In this case, when couriers arrive at the merchant's wrong location for the first time and cannot find the merchant, they would call the merchant and find out the real place and then go there to pick up the order. The merchant may also actively call the couriers to go to the real location. They would claim they recently moved and did not update their locations in the platform yet. Next time, these couriers may directly go to this real location when assigned with an order to pick up from the same merchant. (ii) *Collusive attack.* An adversarial merchant colludes with couriers by providing some monetary incentive to the couriers, so they can work together against the platforms even though the platform may ask the couriers to report the wrong locations[2, 15]. This kind of frauds persists because most of the couriers are part-time with little incentive to report errors given many platforms currently do not provide any monetary incentive to motivate couriers to report. (iii) *Collective attack.* Some adversarial merchants close to each other register the wrong yet close locations collectively.

2.2.3 Impact of Location Frauds. Merchants with location frauds make both new and experienced couriers take detours of several kilometers to pick up orders, which impacts the platform's efficiency significantly. It also compromises the platform's credibility on the delivery time estimation, which lead to poor customer experience. Moreover, location frauds are not rare. Fig. 2 (b-d) show the scale and impacts of location errors based on real-world data.

2.3 Opportunities

2.3.1 Key Idea: Correlation between Travel Time and Distance. Our solution to detect location frauds is based on a simple yet effective idea. *The travel time between a pair of consecutively visited merchants should be proportional to the travel distance between them for large-scale merchant pairs.* Based on this correlation, we can estimate the travel distances between any two merchants using the travel time.

2.3.2 From Travel Distances to Location Frauds. We compare two distances: the travel distance estimated by arrival time captured by platform infrastructure (detailed in Sec.3) for order status, and the travel distance estimated by registered locations provided by merchants. If their difference is significant, at least one of them is wrong. Compared to the distance obtained by registered locations, the distance obtained by the arrival time is more likely to be correct because the platform invests significant effort (Sec.3) to collect arrival time for accounting. Thus, in our detection, we (1) use *arrival time* from order status data (as in Tab. 1) to infer *travel time* between any two merchants; (2) use *travel time* between any two merchants to infer *travel distance* between any two merchants; (3) use *travel distance* from one merchant to nearby merchants to cross-validate if this merchant has a suspicious registered location, i.e., *location fraud.* An updated example of Fig. 2(a) is given in Fig. 5. Based on three merchants A, B, and C with their registered locations and travel time between them, we found that (1) the travel time between A and B is much shorter than it should be; (2) the travel time between A and C is much longer than it should be; (3) the travel time between B and C is similar to it should be. As a result, we infer the registered location of Merchant A could be wrong. By a field study, we found out its real location at the hollow circle, which explains the abnormal travel time between them.

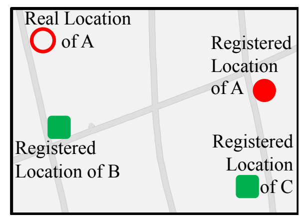


Fig. 5. Distance Vs. Time

2.3.3 Foundation of Detection: Data Volume and Merchants Density. Although the abnormal travel time could be because of various personal and environmental factors such as couriers' preference rather than location fraud, we assume most carriers take the shortest routes to deliver orders based on previous studies[64]. As a result, our idea (i.e., the correlation between time and distance) is based on large order volumes and dense merchants. As in Fig. 6 and 7, 58% merchants have 300+ orders per month, and 73% of them have 30+ merchants in 5 km in Shanghai. This high density inspires us to cross-validate locations for detection.

2.4 Technical Challenge

The key challenge we aim to address is *the uncertainty of couriers mobility behaviors.* Fig.8 plots the travel distances and the travel time for 1,000 orders. The diagonal is the normal walking speed, i.e., 1 meter per second; the red dotted line is the regressed linear function. Both of them cannot represent the relationship well. The mobility behaviors of couriers, e.g., travel speeds and routes, are affected by many factors, e.g., the remaining time to overdue, and couriers' transportation modes (walk or e-bikes). As a result, even for a fixed merchant pair, the travel time is unstable. We draw the CDF of the standard deviations for travel time between the same

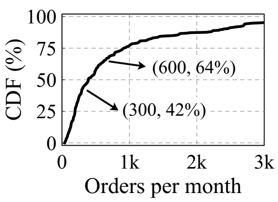


Fig. 6. Orders per Month for Each Merchant

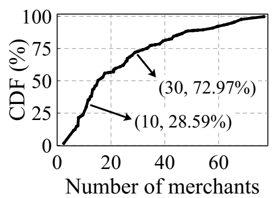


Fig. 7. CDF of Nearby Merchants

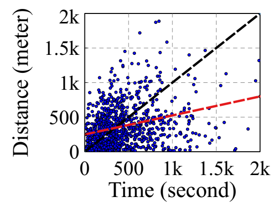


Fig. 8. Scatter Plot of Travel Time and Distance

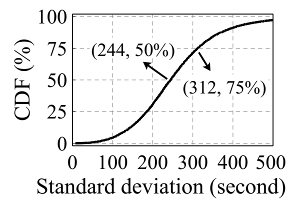


Fig. 9. STD for Travel Time of Merchant Pairs

merchant pairs in Fig. 9. More than 50% of merchant pairs have the travel time variations larger than 244 seconds, accounting for 14% of a typical order delivery time (30min). It brings a significant challenge for our system.

3 SYSTEMS AND DATA

3.1 Order Status Data Collection Systems

Order status data shown in Tab. 1 are significant to the platform because they are used for the platform's new order assignment, and are also shown to customers in real time to improve customers' experiences. While the order status data are collected for the platforms' daily management and accounting, location fraud detection becomes an interesting "side product". In most online delivery platforms, these data are collected manually with lots of errors. In our platform, we deploy both physical and virtual check-in devices to automatically collect the order status data. We introduce these three data collections below.

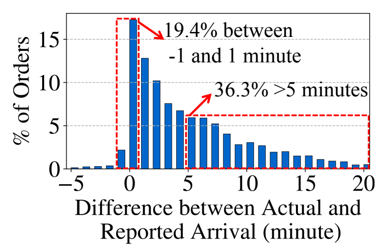


Fig. 10. Inaccurate Reporting

(1) Manual Collection based on Courier Smartphone Only Based on the platform policy, couriers are obligated to report the order status manually during delivery. However, these reports are often inaccurate. We compare the reported arrival time and the actual arrival time (collected from beacon devices) in Fig. 10. It shows that only 19.4% of order reports are accurate, and 36.3% of them have been reported in advance for more than 5 minutes (accounting for 17% of the whole order delivery time). There are multiple incentives for couriers to report early including avoiding overdue responsibility and gaming order scheduling. Couriers also forget to report progress sometimes, which leads to very late reporting. The uncertainty of couriers' behavior brings inaccurate timings.

(2) Automatic Collection based on Physical Check-in Beacon System To automatically obtain the timing of couriers' arrival at and departure from merchants, our team (i.e., the platform) has designed and deployed 3,200 beacon devices in Shanghai. Fig.11 visualizes such a beacon device deployment, where each merchant is equipped with one beacon device. These devices have basic communication capability but no GPS capability due to the energy and cost concerns. They are mailed to or picked up by the merchants, who will register the locations of installed beacon devices, which could be a fraudulent location. Since these deployed beacon devices whose IDs are one-to-one correspondence with merchant IDs, the arrival and departure timings are straightforward and automatic for couriers. A courier's smartphone detects the beacon message constantly broadcast by physical beacon devices when in proximity. Then, couriers' smartphones upload the timestamp and the beacon device's ID to the server, which implies the courier is around the merchant with this beacon device, so an arrival time was recorded for this courier. In our designs, we add energy management schemes to ensure beacon devices can

survive more than two years without changing batteries. Additional security mechanisms have been also designed to avoid free-ride. However, the large-scale physical beacon system is labor-intensive and cost-ineffective (10 USD per device).

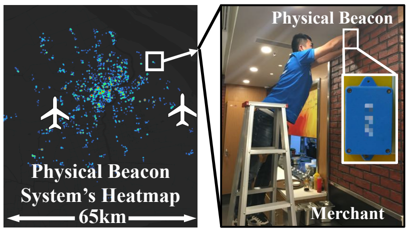


Fig. 11. Physical Beacon System Deployment

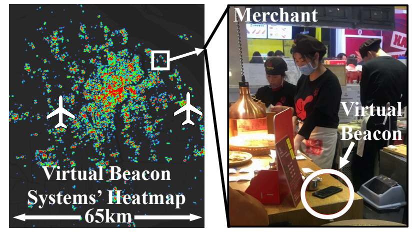


Fig. 12. Virtual Beacon System Deployment

(3) Automatic Collection based on Virtual Check-in Beacon System To reduce the cost, our team also deployed a virtual beacon system that is part of the smartphone app of merchants. The app is used by merchants to manage the orders received on the platform. As in Fig. 12, the virtual beacon broadcast module is added to the merchants' app under merchants' consent in Shanghai. When a courier approaches a merchant, her smartphone will receive the virtual beacon messages generated by the merchant's smartphones. The mechanism of timing collection is the same as the physical beacon device we mentioned above. The virtual beacon message does not include continuous GPS tracking to protect the privacy of merchants but has a dynamic ID, so the merchant can be mapped based on merchant IDs on the server for couriers' arrival detection.

3.2 Temporal Data Collection

Based on the data collection system, we introduce the collection of temporal data T1 to T3 as follows. Note that all the three data collection infrastructures are in use simultaneously. Physical and virtual beacon devices are in the testing stages for replacing couriers' manual report of order status.

- **T1: Manual Timing Data from Couriers' Input (Man)** In manual data collection, the platform requires couriers to mark their arrival and departure at each merchant in the couriers' app. The timestamps will be uploaded to the platform server at no additional hardware cost.
- **T2: Physical Beacon Timing Data (Phy)** Based on the records (uploaded by courier smartphones) with beacon devices (i.e., the timestamps and beacon ID), we obtain the exact timing when a courier passes through a merchant automatically. When a courier passes a merchant, she may receive multiple beacon messages. We select the first and last timestamps from the same beacon to the same courier. The first timestamp is regarded as the courier's arrival and the last one as the departure.
- **T3: Virtual Beacon Timing Data (Vir)** The way of obtaining the timing from virtual beacon devices is the same as the physical beacon devices. To effectively compare the performance, we only use the data of virtual beacon devices whose merchants also have physical beacon devices in the evaluation. Note that some merchants use PC to manage orders, so we cannot use their smartphones as virtual devices. Some merchants also do not want to participate, so we do not consider them.

3.3 Spatial Data Collection

There are three approaches (L1-L3) to obtain the couriers' locations given the arrival/departure timing with uncertainties. Note that in our setting, given the timing of arrival, a courier's location is the merchant location. The ground truth of the merchant location (L3), e.g., the real location, may not be the location obtained by either of two approaches.

- **L1: GPS Locations of Couriers (GPS)** A GPS record is obtained when a courier reports arrival or departure on her app (i.e., manual timing collection), or detected a beacon message (i.e., automatically timing collection with physical or virtual beacon devices). However, GPS in urban areas is often inaccurate[56], and some couriers have GPS modification software to change their GPS locations to game the platform's accounting system[30].
- **L2: Registered Locations of Merchants (Reg)** These locations are obtained by matching the merchant IDs uploaded by couriers (either from manual report or beacon devices detected) with the merchant IDs in the databases. Each merchant ID has a unique location registered by the associated merchant.
- **L3: Ground Truth of Merchant Locations** Our platform has a field study team to periodically visit some merchants with abnormal orders. During this process, the ground truth of erroneous locations is collected manually but with a high labor cost. The ground truth gives the real locations of the merchants in Tab. 2.

Table 2. Ground Truth Format and Example

Field	Value
Merchant ID	M001
Registered Location	116.418065, 39.916998
Real Location	116.431181, 39.914571

Based on the ground truth, there are over 93 merchants of all 3,200 merchants with both physical and virtual beacon devices in Shanghai with significant location errors, accounting for 2.9% of all 3,200 merchants. The location distribution of erroneous merchants is basically consistent with the correct ones, covering both urban and suburban areas. We observed a typical fraud that some merchants intentionally claim to be located in a large shopping mall to attract more online consumers because merchants in malls implies better qualities. The common unintentional error is that the merchant registered her address with similar but wrong mall or street names by mistakes. Because two addresses with similar names may be far away, this kind of mistakes leads to server errors in practice. Although the ratio of identified frauds among all merchants is small, the total amount is significant due to a large number of merchants.

4 ALWAES DESIGN

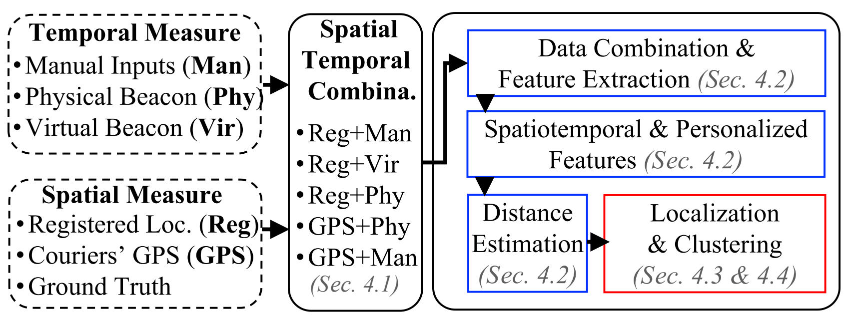


Fig. 13. Overview of ALWAES

In this section, we present the design of ALWAES system that explores these temporal and spatial data to identify the location frauds and infer the real locations. Fig. 13 gives an overview of ALWAES. With the temporal and spatial data collected (Sec.3), ALWAES explores different combinations on spatiotemporal measurements that lead to different tradeoffs on performance and cost (Sec.4.1). For any of these combinations, we can train a travel distance model that helps to estimate the travel distance between nearby merchants (Sec.4.2). We then localize the merchants based on these distances and compare them with their registered locations to detect frauds (Sec.4.3). As couriers usually travel within small areas, we partition the merchants into small clusters that are geographically close to each other in the first step (Sec.4.4). This step can filter out some noise and simplify the merchant localization process.

4.1 Spatiotemporal Measurement

To evaluate the performance of ALWAES in different settings, we generate five measurement combinations in the spatiotemporal dimensions mentioned above, as shown in Fig. 14, and explore the tradeoffs of them in Tab. 3.

Table 3. Measurement Strategy Comparison

	Implemet.	Performa.	Cost	Density
①Reg+Man	Manual	Inaccurate	Cheap	Sparse
②Reg+Vir	Automatic	Accurate	Cheap	Medium
③Reg+Phy	Automatic	Accurate	Expensive	Dense
④GPS+Phy	Automatic	Accurate	Expensive	Dense
⑤GPS+Man	Manual	Inaccurate	Cheap	Sparse

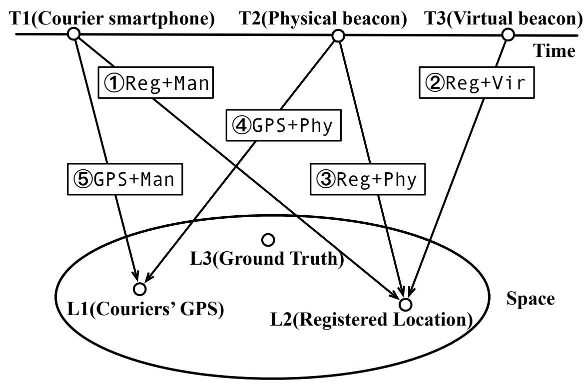


Fig. 14. Demonstration of Measurement Combinations

① **Reg+Man.** It utilizes the **Manual** input order status records (**T1**) and merchant **Registered** locations (**L2**), which may introduce manual errors of couriers (e.g., early or late reports) and thus the timings can be inaccurate.

② **Reg+Vir.** It utilizes the records with **Virtual** beacon devices (**T3**) and merchant **Registered** locations (**L2**). Couriers usually pass by multiple merchants when picking up orders from different merchants, so the timing records with beacon devices are denser than Reg+Man. As a result of automatic recording, the timings are accurate.

③ **Reg+Phy.** It utilizes the records with **Physical** beacon devices (**T2**) and merchant **Registered** locations (**L2**). For the same reason as Reg+Vir, the timings are accurate. Compared with Reg+Vir where virtual beacon devices depend on merchants' smartphones, the density of physical beacon devices (i.e. Reg+Phy) is potentially higher because all the merchants in the evaluation have physical beacon devices, but virtual beacon devices only exist in the merchants that use smartphones to manage orders. The cost of Reg+Phy is highest given the hardware cost and labor-intensive deployment, but no maintenance cost is needed after deployment.

④ **GPS+Phy.** It utilizes recorded timings with **Physical** beacon devices (**T2**) and couriers' **GPS** locations (**L1**). Similar to Reg+Phy, the data collection of GPS+Phy is expensive due to physical beacon devices. Further, GPS+Phy needs GPS locations as extra information compared to Reg+Phy. Due to security and privacy concerns, GPS is not an attractive solution in many crowdsourcing applications. As a result, Reg+Phy is easier to implement than GPS+Phy.

⑤ **GPS+Man.** It utilizes couriers' **GPS** locations (**L1**) and **Manual** input timings (**T1**). Similar to Reg+Man with manual input, the timings of GPS+Man are less accurate than GPS+Phy. While it is cheaper to collect than GPS+Phy, it needs GPS compared with Reg+Man, which leads to privacy issues and can be manipulated[56] or even attacked, e.g., by GPS spoofing[30], resulting in severe location fraud issues.

4.2 Distance Estimator

Based on these five measurement strategies, we show how to estimate the travel distances between merchants based on the travel time and related features. Note that our estimator works with any of these five measurement data as input, given their inherent homogeneous spatiotemporal measurement nature. We evaluate the performance of our estimators with different measurement data in Sec.5.

4.2.1 Feature Extraction. The travel distances are related to different types of features, e.g., temporal, spatial, and personalized. *Temporal features* reflect the primary time-related information, including the travel time, peak hours, the latest delivery time, etc. *Spatial features* reflect the geographical factors, including the number of orders per unit area, delivery difficulty, etc. *Personalized features* reflect the courier's behaviors and preferences such as the number of carried orders, the number of overdue orders in history, and history travel distance.

To identify the most impacted ones, we adopt a series of standard feature selection algorithms in literature [10, 29, 31]. We first remove the features with low variances to reduce the number of features to avoid "curse of dimensionality" We use mean decrease accuracy[32] (i.e., the decrease in the model accuracy from permuting the values in each feature) to score the feature importance. These scores weighted equally in assessments. As a result, we keep the 32 most impacted features. More details of the feature selection techniques are given in [10, 29, 31]. The full list of features is given in Tab. 4. Given these features, we estimate the distances between merchants with a distance model.

Table 4. Feature List

Temporal: travel time, peak hours, longest delivery time, minimal estimated travel time, historical and averaged delivery time, team history deliver time, to promise delivery time, time slot, merchant history cook time, merchant history time of picking up goods, merchant history deliver time, grid average cook time, grid average time of picking up goods, grid average deliver time
Spatial: delivery difficulty, rate of delayed products, merchant popularity, number of orders and finished orders in the last 30min, shopping mall ID, number of orders per unit area, grid delivery difficulty, grid history deliver time per kilometer, the density of couriers in the area
Personalized: number of carried orders, courier's passion, number of overdue orders, accept orders time of the courier, average history travel distance, courier grade

4.2.2 Distance Model and Methodology Choice. With these 32 carefully selected features, we build a distance model based on the Gradient Boosting Decision Tree (GBDT) [27], a widely adopted machine learning algorithm. The main reason we build our distance model based on GBDT is that compared with models such as logistic regression and SVM, GBDT is able to learn the nonlinear relationship between features and labels, which adapts to our problem. As a decision tree model, GBDT can explain the feature importance, which sheds the light on the couriers' mobility behavior patterns. We use Mean Absolute Error(MAE) as the loss function in optimizations.

Map Travel Distance as Labels. We use the travel distances calculated in the map app, called "*map travel distance*" $\mathbf{d_{map}}(\mathbf{p}, \mathbf{q})$ as labels. Considering the different travel modes for different travel distances (e.g., on foot or e-bike), we train two models: one for on-foot with shorter distances (e.g. $< 100m$), and the other for e-bikes with longer distances (e.g. $\geq 100m$). We set d_{bike} as the threshold for long distance. As couriers in China rarely drive a car to deliver orders, we did not consider the third model for vehicles.

Model Travel Distance as Output. The travel distance obtained by our model between two merchants p and q is called the "*model travel distance*" $\mathbf{d_{model}}(\mathbf{p}, \mathbf{q})$.

4.3 Merchant Localization

Given the model travel distances between merchants in the previous subsection, we localize each merchant and check whether it deviates from the registered location significantly.

4.3.1 Setting. For a merchant with a registered location $\mathbf{q}^*$, called "*target merchant*" (the registered location but could be fraudulent), suppose we are given a set of nearby merchants $\mathbf{P} = \{p_i, i = 1...N\}$ called "*reference merchants*". $d_{\text{model}}(p_i, q^*)$ is the model travel distance from q^* to p_i from the distance estimation model. Our goal is to compute a location $\hat{\mathbf{q}}$ based on the locations of these reference merchants $\mathbf{P}$ and $d_{\text{model}}(p_i, q^*)$. We then compare this computed $\hat{q}$ with q^* to see if they differ significantly, e.g., >200m. This procedure is conducted for each merchant in turn. In other words, every merchant will be the target merchant for one time and be the reference merchant for its nearby ones.

Specifically, let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ represent the *road network* where V is the set of *road intersections* and $E = \{e_{v_1,v_2} : v_1, v_2 \in V\}$ is the set of *road segments*. Given two merchants p_1 and p_2, $d_{\text{map}}(p_1, p_2)$ is the map travel distance in between. In Fig.15, $d_{\text{map}}(p_1, p_2)$ is equal to $d_{\text{map}}(p_1, v_1) + |e_{v_1,v_2}| + d_{\text{map}}(v_2, p_2)$ where $|e_{v_1,v_2}|$ is the road length. Here $d_{\text{map}}(p_1, v_1)$ is the edge length $|e_{p_1,v_1}|$. In this paper, we assume the shortest route when computing the map travel distance because couriers seldom take longer routes. Thus in the example of Fig.15, $d_{\text{map}}(p_1, p_2)$ is for the route (p_1, v_1, v_2, p_2) but not (p_1, v_4, v_3, p_2).

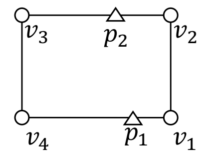


Fig. 15. Road Network

Given a set of reference merchants ($P = \{p_i\}$) and model travel distances $d_{\text{model}}(p_i, q^*)$, the merchant localization problem is to find a $\hat{q}$ that minimizes the difference between the model travel distances $d_{\text{model}}(p_i, q^*))$ and the map travel distances $d_{\text{map}}(p_i, \hat{q})$ for all $p_i \in P$, i.e.,

$$\hat{q} = \arg\min_{q} \sum_{p_i \in P} (d_{\text{map}}(p_i, q) - d_{\text{model}}(p_i, q^*))^2 \tag{1}$$

This optimal $\hat{q}$ should be close to the registered location q^*. If not, we identify it as a suspicious merchant for manually verification. Considering our problem is based on the travel distances that merchants are constrained on the road network, we name it as "Graph-based Multilateration".

4.3.2 Segment Search Algorithm. To find the solution of the Graph-based Multilateration problem, i.e., the $\hat{q}$ in Eq. 1, we design a segment search algorithm as shown in Algo. 1. The basic idea is to scan all the road segments, and for each road segment, we search the optimal location. The solution in Algo. 1 is globally optimal in polynomial time, which makes it scalable for periodical large-scale detection, e.g., once per day or per week, to reduce or minimize the delay to detect a fraud. We omit the detailed proof due to the space limitation but show some key ideas. In short, Eq. 2 is a convex function about $x(d(u, x))$, and thus it can be directly solved by Newton method. The iterations are invoked for each edge e, so the computation complexity of the algorithm is $O(|E|)$, i.e., polynomial.

Algorithm 1: Merchant Localization

Input: Road network $G = (V, E)$; Reference merchants $P = \{p_i\}$; Model travel distance $d_{\text{model}}(p_i, q^*)$
Output: A location $\hat{q}$

min_g $\leftarrow \infty$
for $\forall e_{u,v} \in E$ **do**

$$g(q) = \sum_{p_i \in P} \min\{(d_{\text{map}}(p_i, u) + d_{\text{map}}(u, q) - d_{\text{model}}(p_i, q^*))^2, (d_{\text{map}}(p_i, v) + d_{\text{map}}(v, q) - d_{\text{model}}(p_i, q^*))^2\} \tag{2}$$

  q= $\arg\min_{q \in e_{u,v}} g(q)$
  if *g(q) < min_g* **then**
    min_g=$g(q)$, $\hat{q} = q$

4.4 Merchants Clustering

Intuitively, only nearby merchants are helpful to mutually verify the correctness for each other. Distant merchants have few courier's travels in between and provide little information. More importantly, long travels may introduce greater variation noise to the data. Therefore, we run the distance model and localization algorithm only locally. This is achieved by clustering the merchants to small subsets and the distance estimation and localization are conducted within each cluster. All merchants are clustered based on the following intuitions.

- The urban and suburban areas should be separated.
- The inner and inter-cluster distances should be significantly different. Our empirical experiences show that the typical inner-cluster distance is 2KM; whereas the typical inter-cluster distance is at least 5KM.
- The covered area of a cluster is limited by 2KM.

Given these intuitions, we cluster the merchants based on the hierarchical clustering algorithm BIRCH [9, 66], a widely-adopted clustering algorithm. We start from single-node clusters, iteratively find the pair of clusters with the least inter-cluster distance, and merge them if they are too close. When all iterations finish, we merge the clusters of fewer than 4 merchants to nearby clusters because three points are needed to localize the fourth point.

5 EVALUATION

5.1 Methodology

5.1.1 Baselines. In addition to the five different ALWAES measurements, i.e., Reg+Man, Reg+Vir, Reg+Phy, GPS+Phy and GPS+Man, we also implement ALWAES with three kinds of alternative technical components, i.e., clustering, distance estimation, and localization, summarized in Tab. 5. Note that for fairness, these baselines are all implemented based on the same dataset as Reg+Phy.

Table 5. Algorithms in Comparisons

Components	Clustering	Distance Model	Localization
Linear-based	✓	Linear-based	✓
Trajectory-based	✓	Trajectory-based	✓
MDS-based	✓	✓	MDS-based
ALWAES w/o Cluster	None	✓	✓
ALWAES	✓	✓	✓

- **Linear-based Algorithm** calculates the travel distances by linear regression with the travel time [13].
- **Trajectory-based Algorithm** computes the travel distances based on the courier's GPS data (under their agreements) [69]. We track the couriers' GPS every 20 seconds, which is for evaluation only because the platform does not need their continuous GPS in operations.
- **MDS-based Algorithm** differs from ALWAES at the localization stage. MDS [17, 65] has been widely adopted in localization. It requires the pairwise distances between every pair of merchants in computation. For those without distance input, a multi-hop shortest path distance is used instead (We use the Floyd-Warshall algorithm).
- **ALWAES w/o Cluster:** To show the impact of clustering, we also implement ALWAES without the cluster component so all merchants and data are mixed to train a single travel distance model, and then merchants are localized.

5.1.2 Evaluation Metrics.

- **Location Fraud Detection:** Because it is a binary classification problem, i.e., a merchant's location is fraudulent or not, we adopt the Area Under The Curve (AUC) and Recall.[25]. AUC is the probability that a classifier will rank a randomly chosen positive instance higher than a negative one. AUC=0.5 is a lower bound, which can be obtained by a random guess. In our framework, we pay more attention to the missing detection (false negative) than unnecessary checks (false positive). It is because the platform will double-check the fraud manually according to the detection result of ALWAES to avoid mistaking merchants as frauds. As a result, we choose the threshold to maximize Recall (instead of Precision) within the manual detection capability (detecting 25% of merchants in this paper).

- **Distance Estimator:** We adopt MAE, Root Mean Squared Error (RMSE), and Median Absolute Error (MedAE) to evaluate the performance of different algorithms.
- **Merchant Localization:** The localization error is measured by Euclidean distance between real location q (i.e., ground truth) and computed location $\hat{q}$.

5.2 Overall Results in Validation Phase

Fig. 16 shows the comparison of AUC and Recall between ALWAES and four baselines. Different versions of ALWAES from three measurement infrastructures (i.e., Reg+Man, Reg+Vir, Reg+Phy) outperform the four baseline algorithms. The four baseline algorithms use the same inputs of Reg+Phy. And Reg+Phy outperforms GPS+Man but not GPS+Phy.

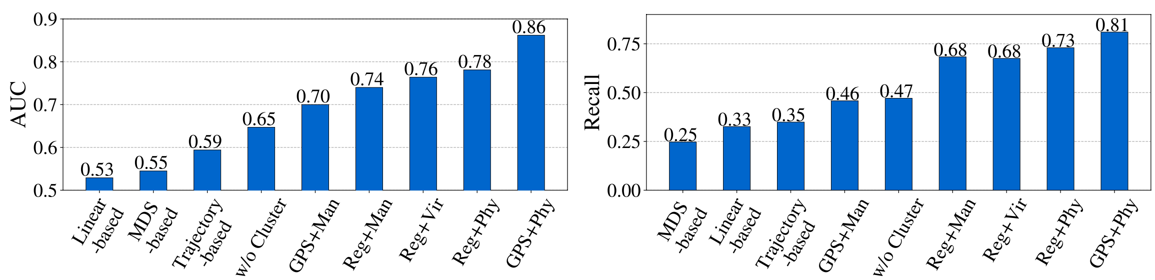


Fig. 16. AUC and Recall Comparison

5.2.1 Comparison with Baselines. We found that all the other four baselines (Linear-based, Trajectory-based, MDS-based, and w/o C) have AUC around 0.5 to 0.6. For the MDS-based algorithm, its poor performance is mainly due to the lacking of distance data on some pairs. In our work, each cluster has 54 merchants on average, while 37% of merchants pairs have no travel data. For this, the MDS-based algorithm [65] has to fill the distance matrix with estimated shortest paths, which brings great errors. By comparing these baselines, we found that the distance model component has the biggest impact, followed by the localization and clustering component.

5.2.2 Comparison between Measurement Combination. We found only GPS+Phy is better than Reg+Phy, which is expected given the advanced inputs of GPS+Phy from both beacon devices for timing and courier smartphone's GPS for locations. Reg+Man outperforming GPS+Man is because GPS+Man suffers from GPS errors in the urban area. Localizing merchants based on instantaneous GPS are more affected by the uncertainty of couriers reporting than getting travel time which may eliminate some uncertainties by timings subtraction (a certain courier may have similar behavior, e.g. always reporting in advance). Reg+Vir outperforms Reg+Man because Reg+Man suffers from unreliable manual timing inputs; Reg+Phy outperforms Reg+Vir because Reg+Vir suffers from the low density of merchants with virtual beacon devices. Because only 87% of 3,200 merchants have virtual beacon devices but all of them have physical beacon devices. The fundamental trade-offs between timing, locations, hardware, and software are in the lessons learned of Sec. 6.

5.2.3 Failure Scenarios. We found 25 merchants that are fraud locations but ALWAES fails to identify. We make further investigations on these false-negative cases and there are two cases. *(1) Collective Fraud.* Some fraud merchants may be close to each other, intentionally or coincidentally. Based on our results, we found that a set of collective frauds closer to each other will make them hard to be detected and even mislead the model to believe some correct merchants are erroneous if the density of fraud merchants is higher than the correct merchants. There are 9 collective fraud merchants in our evaluation. But as indicated by our ground truth, there are only 2.9% of fraud merchants, which makes our model performs well in practice. *(2) Non-unique Localization.* The topology of nodes in a cluster is crucial for the localization. When the reference merchants are very close in distance and have a symmetric topology, they are unable to localize merchants uniquely and large errors are introduced. But this case is also rare in the urban areas because of the high merchant density. In order to enhance the merchants' experience, our platform has a field study team to periodically visit some merchants with abnormal orders, and it is also the remedial measurement against the failure scenarios. When a fraud merchant is found, our team will visit his/her neighbors to avoid collective fraud. Meanwhile, the field study team will double-check the ALWAES tagged fraud merchants to avoid incorrect tagging. The platform also has manual customer service as an appeal channel for merchants to correct erroneous tag.

5.3 Results in Real-world Testing Phase

We deploy a well-trained version of ALWAES in the real-world platform operation in a one-week testing phase. In this phase, we identified new erroneous merchants and misidentified some legitimate merchants. We also report the impacts of ALWAES courier detour distance reduction.

5.3.1 New Fraud Detection. We identify eight additional fraud merchants that are not identified by the platform initially but verified by our field study. We choose to manually verify these potential fraud merchants detected by our model but not in the ground truth because these merchants have rich route records from reference merchants that unrealistically seem far away according to the reported locations.

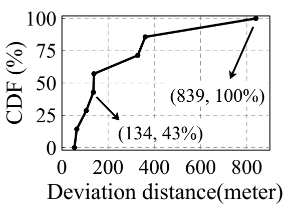


Fig. 17. Actual Deviation

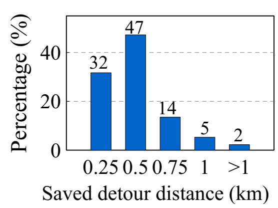


Fig. 18. Saved Detour Dis.

Table 6. The Results of the Distance Estimator

	Linear-based	Trajectory-based	ALWAES
MAE(m)	196.56	404.31	77.31
RMSE(m)	244.57	855.63	111.72
MedAE(m)	175.35	164.20	57.80

5.3.2 Courier Detour Distance Reduction. Fig. 17 shows the CDF of the actual deviation distance of detected erroneous merchants from their real locations. Half of the merchants deviate more than 136 meters and the maximum deviation distance is up to 839 meters. To show the effectiveness of ALWAES on saving the unnecessary detours of couriers, Fig. 18 shows the distribution of the saved detour distances by ALWAES. We found that 20% of the saved detours are more than 500 meters, and some even exceed 1 km. This is equivalent to 3,846 hours of saved delivery time for couriers in total based on the average travel distance of couriers.

5.4 In-depth Results on Components

We evaluate the components of ALWAES based on Reg+Phy.

5.4.1 Distance Estimator Component. Tab. 6 compares the distance estimations of ALWAES and other two distance estimator algorithms. It shows that ALWAES has only 77.31 meters MAE, while the MAE of Linear-based and Trajectory-based is 196.56 meters and 404.31 meters, respectively. The MAE is significantly reduced by 60.66% and 80.87%. These good results are mainly due to the advantage of the 32 different features and the machine learning algorithm we developed. It is interesting to observe that the Trajectory-based algorithm does not perform well either. This is mainly because in an urban area, the GPS signals drift largely and severely and even the GPS-based trajectories cannot well measure the travel distances between merchants.

We use the “all but X” technique[26] to explain the feature importance for an ablation, removing one feature then check the MAE against the original model. Fig. 19 shows the most important features and MAE without them. Travel time is the key input. Without travel time, the MAE increases from 77 to 315 meters. The order related features, e.g., the rate of merchant’s delayed products (i.e., an order was prepared late), also help our distance estimation.

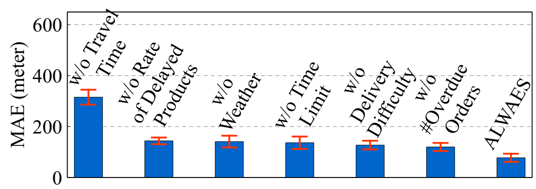


Fig. 19. Feature Importance

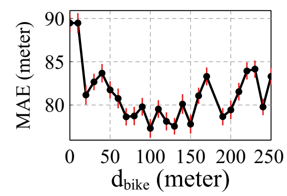


Fig. 20. Impact of Different d_{bike}

The e-bike threshold d_{bike} is used for distinguishing the different travel modes based on different travel distances (i.e. on foot or e-bike) (detailed in Sec. 4.2.2). Fig. 20 demonstrates how the MAE of Distance Estimator varies with d_{bike}. We can see that it performs relatively stable and well when d_{bike} is around 100 meter. The d_{bike} equal to 0 meter means considering only one travel mode. It shows that considering two modes improve the performance of Distance Estimator. It is because the couriers in China rarely deliver long-distance orders on foot considering time limitations. As a result, it is necessary to generalize the finding to other cities in China that couriers have different travel modes. And as they rarely drive a car to deliver orders, we did not consider the third model for vehicles.

Impact of Data Length, Courier Count, Merchant Density. As shown in Fig. 21 (a) and (b), our model is insensitive to data length and couriers scale, while Linear-based and Trajectory-based algorithms degrade dramatically when the evaluation scale increases. This is because the simple linear regression model is inadequate for handling data noise in a large dataset. Fig. 21 (c) reveals a significant improvement of our model when the merchant density increases. This is because more reference merchants bring more training data, which makes our distance model more accurate.

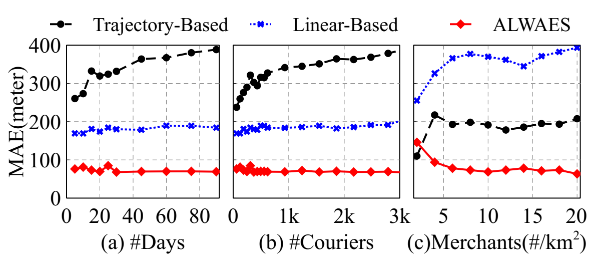


Fig. 21. Impact on Distance Estimator

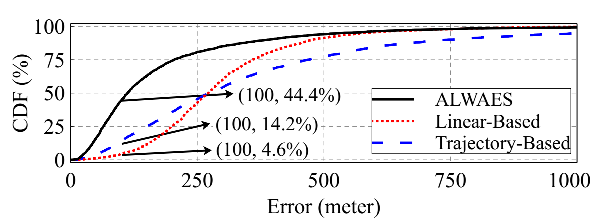


Fig. 22. CDF of Localization Error

5.4.2 Localization Component. We plot the CDF of localization errors in Fig. 22. It shows that 44.4% of the ALWAES errors are lower than 100m, while that of the Trajectory-based algorithm is 14.2% and the Linear-based is 4.6%.

Impact of Data Size, Courier Count, Merchant Density. The localization error for merchant density is in Fig. 23 (c). For ALWAES, the localization error continuously decreases with more nearby merchants. More nearby merchants bring more reference nodes for localization, and hence the performance improves. When 10 merchants per square kilometer are as reference merchants, the localization errors are dropped to 100 meters. For the

Linear-based and Trajectory-based, the benefit of more reference nodes is weakened by the high noise of data and degraded distance estimator, resulting in a worse performance with higher merchant densities. The impact of the datasets size and the number of couriers on the Localization is shown in Fig. 23(a) and Fig. 23(b) with similar observations.

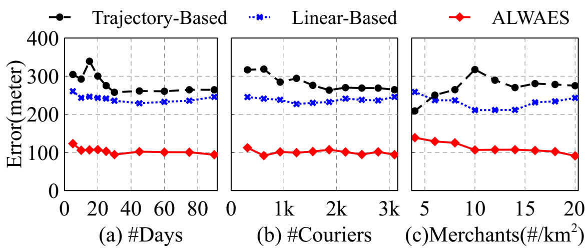


Fig. 23. Impact on Localization Component

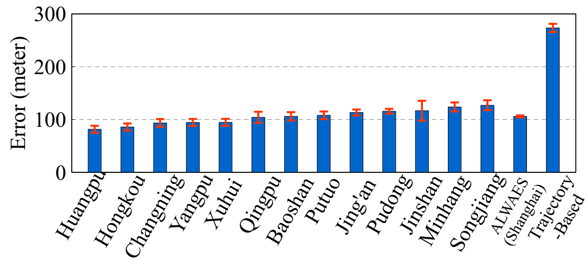


Fig. 24. Localization error in different districts

Generalization in Different Locations. The couriers' and merchants' behaviors may differ in different locations. With the limited resources in practice (e.g., ground truth data collection), we are only able to evaluate our work in only one platform at a city scale (6 million residents and 660 square kilometers of the urban area; 13 million residents and 3768 square kilometers of the suburb area). In order to analyze the generalization in different locations, we evaluate the ALWAES performance among different districts of Shanghai, which stand for various behaviors to some degree such as downtown, suburb, industrial parks. The result is shown in Fig. 24. The three districts with poor performance (Songjiang District, Minhang District, and Jinshan District) are all suburb districts (1.6 million residents and 521 square kilometers of area per district on average) that have fewer couriers and merchants than urban downtown districts. It indicates that ALWAES may work better in downtown regions of cities with a similar merchants' scale to Shanghai. But ALWAES can still provide better performance than baselines in suburb areas.

The detection threshold is used for comparing the difference between the registered location (q^*) and the computed location ($\hat{q}$) to decide if a registered location is a potential fraud or not. To investigate its impact, we explore recall over different thresholds in Fig 25. We set the threshold as 0.2 km based on the manual double-check capability and the recall is 0.73.

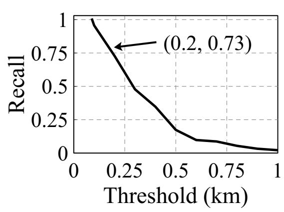


Fig. 25. Recall vs. Threshold

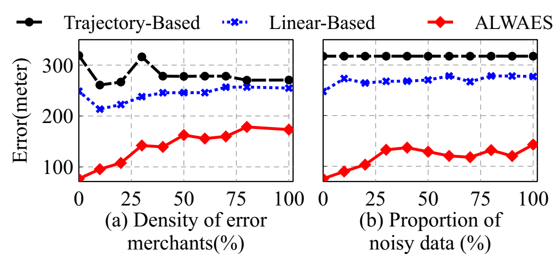


Fig. 26. Robustness for Localization

5.5 Robustness

We evaluate ALWAES's robustness against the density of erroneous merchants (i.e. collective attack) and the noisy timing data on the physical beacon devices' input. The robustness of noisy timing data can also be treated as the robustness against collusive couriers' density (i.e. Collusive attack). Because ALWAES do not use the couriers' incentive reports of merchants' location. The collusive couriers' count (i.e. **collusive attack**) makes no difference on ALWAES. However, if the fraud merchants know how our system works, they may ask the collusive couriers to report order status manually in the fraud locations. In this case, collusive couriers bring noisy timing manually to our system. Because we implement ALWAES in a real-world platform with a fixed portion (i.e., density) of erroneous merchants, we cannot directly evaluate the impact of different density of erroneous merchants on ALWAES. Thus, we conduct a data-driven emulation to study this impact.

5.5.1 *Impact of Erroneous Merchant Densities.*

We randomly select some merchants and manually drift their locations from 100m to 300m. Fig. 26(a) shows the results of localization errors in different proportions of erroneous merchants. These errors continuously increase as more merchants are erroneous. When 50% of merchants are erroneous, the localization errors are up to 180m. This is already significant, as their location drift is 200m on average. For others, their errors are more than 200m and can hardly be even worse. Thus, we find ALWAES can be functional (error less than 100m) when no more than 20% of merchants are erroneous.

5.5.2 *Impact of Physical Beacon Devices' Timing Error.*

We assume the physical beacon devices' timing inputs are correct. To evaluate the performance with different timing errors, we randomly select data in different proportions and manually add noise to the travel time feature from -3 mins to 3 mins. Fig.26(b) shows the results with different proportions of the noisy travel timing data. We can see that ALWAES can tolerate 20% of input errors.

6 DISCUSSIONS

6.1 Lesson Learned

Lesson Learned 1: Fundamental Tradeoff between Timing, Locations, Hardware, and Software. By comparing the tradeoff between different measurement infrastructures in Fig. 14 and their performance in Fig. 16, we found that:
• Without any physical or virtual beacon devices for explicit contextual timing, the implicit inaccurate timing for order status update is helpful when using either GPS or reported merchant as locations, i.e., GPS+Man outperforms first four baselines in Fig. 16 but not Reg+Man because GPS drifting makes GPS+Man inferior to Reg+Man.
• With the help of physical beacon devices, Reg+Phy employs the explicit timing from physical beacon devices only and outperforms all the other approaches except the cross infrastructure solution GPS+Phy that takes the advantages of both GPS and beacon devices together.
• Considering the limitations of Reg+Phy which relies on the expensive deployment of physical devices, a virtual beacon system (employs the smartphones of merchants) with a software-based solution Reg+Vir can achieve similar performance by 87.2% of all merchants. Note that Reg+Vir does not require a merchant to install new software but only adding a module to their existing app, which is practical in the real world but needs a high penetration rate for a network effect.

Lesson Learned 2: System's Robustness Against Attack. Because ALWAES do not use the couriers' incentive reports of merchants' location. The count of collusive couriers who do not report merchants' fraud (i.e. **collusive attack**) has no influence on ALWAES. As for worse attacks that collusive couriers bring noisy timing or merchants collective attack, we found that our system is robust to erroneous merchants densities (up to 20%) and Physical beacon devices' timing error (up to 20%) as in Fig. 26. So as long as no more than 20% of merchants providing fraudulent locations, ALWAES is able to detect these fraudulent locations given a relatively correct beacon timing (within 20% noise).

Lesson Learned 3: Component and Feature Performance for Mobility Uncertainty. Under the same data collection infrastructure, our system has shown an advantage to two baselines as shown in Tab. 6 and Fig. 22. We also provide deeper insights in terms of which components are more important (Fig. 16 shows the distance model is the most important) and which features are more important (Fig. 19 shows travel time is the most important).

6.2 Why the Manual Crowdsourcing Approach does not Work

Our platform has asked the couriers to report wrong locations during the delivery, but this is not effective or enforceable in practice. The reasons are as follows.

- Average couriers lack incentives for reporting because their main job is delivering (e.g., 25 orders per day). The feedback process and potential follow-up verification are time-consuming (including taking photos of shops, uploading photos, GPS location and some text description). When couriers arrive at the merchant's location for the first time and make a detour, they may be not willing to report fraud under the risk of remaining orders overdue. And experienced couriers may directly go to the real location when getting an order to pick up from the fraud merchant without checking the authenticity of the merchant registered location.
- When a platform introduces incentives for fraud reporting (1 dollar reward for successful report in our platform), the "eager beavers" phenomenon appears[7, 53]. Some couriers may get more and extra rewards by manipulating or even forging the information, which not only ruins the corrections but also costs more than needed. There are more than 8 million registered couriers nationwide on the platform we are working with in China. Most of them are part-time couriers with potentially untrustworthy given their employment types. It making the design of the incentive mechanism a major challenge.
- Some malicious couriers could even collude with merchants[2, 15]. They can potentially work with merchants by not reporting location fraud. Please see our attack models in Sec. 2.2

It is thus desired to perform fraud correction tasks in a non-incentive and automated manner.

6.3 Limitations

(i) We only consider the erroneous merchants outdoors. The impact of the indoor environment is only considered as the dynamic factor of travel time variances, instead of indoor merchant errors. It is because the indoor errors, e.g., wrong floors in the same building, might not be significant for delivery. Erroneous indoor locations may concern other applications, e.g., indoor navigation, which is out of the scope of this paper. (ii) We envision there is only a minority of erroneous merchants in our system. The results show that ALWAES will degrade obviously when the percentage is higher than 20%. Because the location fraud is caused by human factors, our consideration fits reality. (iii) Given the limited resources, we are only able to evaluate our work in only one platform at a city scale. We envision it also works in other similar online delivery platforms in cities with a similar scale (e.g., New York City and Beijing).

6.4 Generalization and Implication

We believe our methods can be generalized to a broadly of anomaly detection problems in other systems involving spatiotemporal presences. Essentially, ALWAES introduces the idea of anomaly detection based on spatiotemporal measurement. (i) In wireless sensing, ALWAES has the potential to detect the misplaced RFID tags [50] (intentionally or unintentionally) and obtain relative positions[48] by comparing the expected and measured distances between RFID tags. (ii) In location-based social networks (LBSN), ALWAES has the potential to detect fake reviews[45] and malicious accounts[28] based on their spatiotemporal check-in data. (iii) In ride-hailing services such as Uber, ALWAES has the potential to localize the drivers who are with low GPS accuracy in the urban canyon[14] based on drivers pick-up customers update data. Further, our findings and ideas in ALWAES can shed light on similar problems. A major implication lies in the tradeoff between different spatiotemporal data sources because they feature different cost and performance [69]. Accounting data and GPS are widely used given their low cost and great coverage. But due to mobility uncertainty and GPS spoofing, additional hardware (e.g., physical beacon devices) or software (e.g., virtual beacon devices) are needed to guarantee accuracy and coverage, which also introduces extra cost for hardware and user participation.

6.5 Ethics, Privacy and Security

The collection of the couriers' data are under the couriers' agreement as a part of the *Privacy Policy and User Agreement*. In this agreement, the merchants and couriers are notified that the data are being collected and their data will be used to support and improve products and services (including using anonymous data for machine learning or model training)[23, 24]. In the agreement, the merchants and couriers can explicitly opt out the data collection when we ask for the consent. But most of them will not opt out because this data collection provides the platform an opportunity to improve merchant profits and courier income. As discussed in Sec. 5.3 in the paper, ALWAES can help the platform improve its services for saving delivery time to attract more customers to place more orders, which will ultimately benefit both merchants (i.e., high profit from more customers' orders) and couriers (i.e., high income from more delivery tasks and less overdue penalty).

While the analysis of order update data has great potential for social benefits, we must take active actions to protect couriers' and merchants' privacy: (i) Only a few data analysts have access to data and they have signed the data privacy protocol, and there is no data access to the personnel not related to this project. (ii) We have been working on aggregate and anonymous data, and we do not track individual trajectories of couriers in our work. Under merchant owners' agreements, beacon devices only broadcast Bluetooth signals and did not reveal any personal information. The couriers and merchants' IDs are anonymous keys to join different data sets, and any other ID information cannot be tracked or identified in practice. There is no identifiable information (e.g. names and phone numbers) in our dataset to re-identify a real person. (iii) All type of raw data (BLE, GPS trace, etc.) are deleted from the server completely after a preset life-cycle (i.e., 3 months for the current policy), we only keep the statistic information for analysis which are listed in Tab. 4. (iv) There is a strict internal regulatory process to review the dataset before releasing it on the [anonymous] company's official website.

6.6 Data Release

Our data release[2] as introduced in the contribution list will be properly conducted by a strict internal regulatory process to review the data and ensure privacy and security: (i) The identification field will be hashed so that the "JOIN" operation can be used on the released dataset but original IDs cannot be recovered. (ii) We only process the data fields that are useful in this project, and drop others for minimal exposure. (iii) The merchants' stores locations are public information in platforms for customers reference and courier picking up orders. For protecting the privacy of merchants fraud when releasing data, we will follow the state-of-the practice for the data releasing in Alibaba [1], Amazon [5], and Baidu [6]. In particular, we split Shanghai into 350×400 grids, and each grid is a 200 meter × 200 meter square. The merchants' locations are only accurate to the grid. And we remove the location data that is too unique. For example, only one merchant is in a grid. Even though it will degrade the system performance, it is necessary to assure privacy at the expense of some performance when releasing data. And we will try to ensure data validity under the premise of preserving personal privacy. Note that we use accurate data in our real system.

The [anonymous] company has a tracking record to release various data used in research papers for resultant research contests so the research community can build upon the published work. Similar dataset release can be found for Alibaba [1], Amazon [5], and Baidu [6]. We will follow the data format of a previously released data set from an instant delivery platform [19, 64] to protect privacy in the data release.

In our work, we assume all data collected from the virtual and physical beacon devices are correct and the devices are secure. In practice, we designed and implemented an SM3-TOTP algorithm based on RFC 4226 to avoid data spoofing and free-ride, while the designs on beacon devices are out of the scope of this work.

7 RELATED WORK

7.1 Location Correction

Data-driven approaches have been widely used to identify inaccurate locations. Google Maps, Baidu Map, and DianPing provide the users with a feedback web page to report inaccurate locations [51]. Significant resources

[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=107267

have been invested to hire location correction employees, process the data, and make corrections[54]. Applications are also developed to encourage people to upload photos and locations to earn bonuses. A typical app is GOLD panning [3], which collected over 125 million pictures. These methods require either professional skills or significant incentives. A few works are proposed for the location correction problem based on data already collected. The most relevant work is TransLoc[64], which corrects the *couriers' location* by modeling uncertainty in *indoor* travel times in an online delivery platform. However, as it is based on the assumption that all merchants' registered locations are correct, location fraud of merchants will introduce extra errors to the system, which is the main focus of ALWAES. In addition, ALWAES was focused on outdoor merchants and explored the tradeoff of three infrastructures (manual input, virtual and physical beacons); whereas TransLoc[64] was focused on indoor couriers and explored only one kind of infrastructure data, i.e., manually inputted order data.

7.2 Crowdsourcing-based localization

There are many crowdsourcing-based works that require incentives to attract user's participation, e.g., [7, 11, 22, 63]. All of them have to address the issues related to "Eager Beaver", "Lazy Turkers"[22], and even malicious workers[53, 62], which may not be practically in a large-scale real-world setting. Some other work explore location updating approaches based on social media data, platform check-in data, and user feedback data [47, 52, 59, 71, 73]. However, their performances highly depend on the users' education level and regional development, which cannot be largely adopted [47, 61]. GPS and Wi-Fi information is also used to estimate merchants location [34, 36, 40, 43, 49], which require a large scale of infrastructure, and these methods cannot address the intentional location errors. The noisy data problem is also unavoidable because of low sampling and data quality changing according to environmental conditions[12, 42, 46]. Crowdsourcing-based approaches are also proposed for indoor localization, e.g., [33, 44, 55, 57, 65, 67, 70], which employ different types of sensors such as accelerometer, magnetometer, gyroscope, or cameras. These works mainly focus on the localization system errors rather than intentional errors[21, 41].

7.3 Instant Delivery

Many works focus on the study of online instant delivery problems recently, including merchants retrieval[18], delivery time inference[58, 72], route prediction[68], etc. In these approaches, they generally assume the given merchants' locations are correct and ignore the system errors caused by location fraud, which is the main focus of ALWAES.

8 CONCLUSION

We study the fraud location correction problem based on five measurement infrastructures in an online delivery service. ALWAES is designed based on simple timings of courier's arrival and departure at merchants, which can be potentially generalized to various check-in activities. ALWAES partitions the merchants to small clusters, and builds a travel distance model to estimate the travel distances between merchants, and lets the merchants cross validate their locations. Experiment results show that ALWAES outperforms various baselines. We discuss the strengths, limitations, and lessons learned of our system and will release a one-month sample of our dataset to inspire future research on this direction.

ACKNOWLEDGMENTS

This work is partially supported by the National Key R&D Program of China 2018YFB2100300, 2018YFB0803400, and National Natural Science Foundation of China (NSFC) 61925202, 61772046.

REFERENCES

[1] Aliyun. 2020. Welcome to Tianchi Data Sets. https://tianchi.aliyun.com/dataset/.

[2] Jessica Allen. 2020. What is marketplace supplier fraud and how can you stop it? Webpage.

[3] Amap. 2019. GOLD panning. http://gxd.amap.com. [Online; accessed 2-April-2019].

[4] Amazon. 2020. Amzon Prime Now. Webpage.

[5] AWS. 2020. Registry of Open Data on AWS. https://registry.opendata.aws/.

[6] Baidu AI. 2020. Baidu Research Open Acess Dataset. https://ai.baidu.com/broad/introduction.

[7] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 313–322.

[8] BESPOKE. 2018. Why You Need To Get Over Your Snobbery About Dining In Beijing's Shopping Malls. https://bespoketravelcompany.com/urbanedit/why-you-need-to-get-over-your-snobbery-about-eating-in-shopping-malls/. 12-June-2018.

[9] Somar Boubou, AH Abdul Hafez, and Einoshin Suzuki. 2015. Visual impression localization of autonomous robots. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 328–334.

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

[11] Sabine Buchholz and Javier Latorre. 2011. Crowdsourcing preference tests, and how to detect cheating. In *Twelfth Annual Conference of the International Speech Communication Association*.

[12] Chu Cao, Zhidan Liu, Mo Li, Wenqiang Wang, and Zheng Qin. 2018. Walkway discovery from large scale crowdsensing. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 13–24.

[13] Samprit Chatterjee and Ali S Hadi. 2015. *Regression analysis by example*. John Wiley & Sons.

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

[15] DATAVISOR. 2020. Buyer-Seller Collusion. Webpage.

[16] Deliveroo. 2020. Deliveroo. Webpage.

[17] Carmelo Di Franco, Amanda Prorok, Nikolay Atanasov, Benjamin Kempke, Prabal Dutta, Vijay Kumar, and George J Pappas. 2017. Calibration-free network localization using non-line-of-sight ultra-wideband measurements. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 235–246.

[18] Xuetao Ding, Runfeng Zhang, Zhen Mao, Ke Xing, Fangxiao Du, Xingyu Liu, Guoxing Wei, Feifan Yin, Renqing He, and Zhizhao Sun. 2020. 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*. 3026–3034.

[19] Yi Ding, Ling Liu, Yu Yang, Yunhuai Liu, Desheng Zhang, and Tian He. 2021. 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)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/nsdi21/presentation/ding

[20] DoorDash. 2020. DoorDash. Webpage.

[21] Wan Du, Panrong Ton, and Mo Li. 2018. UniLoc: A Unified Mobile Localization Framework Exploiting Scheme Diversity. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 818–829.

[22] Carsten Eickhoff and Arjen P de Vries. 2013. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval* 16, 2 (2013), 121–137.

[23] Eleme. 2021. Privacy Policy and User Agreement for Couriers' APP. Webpage.

[24] Eleme. 2021. Privacy Policy and User Agreement for Merchants' APP. Webpage.

[25] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[26] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 2018. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *arXiv e-prints*, Article arXiv:1801.01489 (Jan 2018), arXiv:1801.01489 pages. arXiv:1801.01489 [stat.ME]

[27] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[28] Qingyuan Gong, Yang Chen, Xinlei He, Zhou Zhuang, Tianyi Wang, Hong Huang, Xin Wang, and Xiaoming Fu. 2018. DeepScan: Exploiting deep learning for malicious account detection in location-based social networks. *IEEE Communications Magazine* 56, 11 (2018), 21–27.

[29] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[30] Zeeshan Haider and Shehzad Khalid. 2016. Survey on effective GPS spoofing countermeasures. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*. IEEE, 573–577.

[31] Mark Andrew Hall. 1999. Correlation-based feature selection for machine learning. (1999).

[32] Hong Han, Xiaoling Guo, and Hua Yu. 2016. Variable selection using mean decrease accuracy and mean decrease gini based on random forest. In *2016 7th ieee international conference on software engineering and service science (icsess)*. IEEE, 219–224.

[33] Zakieh Hashemifar, Kyung Won Lee, Nils Napp, and Karthik Dantu. 2018. Geometric Mapping for Sustained Indoor Autonomy. In *Proceedings of the 1st International Workshop on Internet of People, Assistive Robots and Things*. ACM, 19–24.

[34] Zakieh S Hashemifar, Charuvahan Adhivarahan, Anand Balakrishnan, and Karthik Dantu. 2019. Augmenting Visual SLAM with Wi-Fi Sensing For Indoor Applications. *arXiv preprint arXiv:1903.06687* (2019).

[35] Zhou He, Guanghua Han, TCE Cheng, Bo Fan, and Jichang Dong. 2019. Evolutionary food quality and location strategies for restaurants in competitive online-to-offline food ordering and delivery markets: An agent-based approach. *International Journal of Production Economics* 215 (2019), 61–72.

[36] Shaohan Hu, Lu Su, Hengchang Liu, Hongyan Wang, and Tarek F Abdelzaher. 2015. Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. *ACM Transactions on Sensor Networks (TOSN)* 11, 4 (2015), 55.

[37] InstaCart. 2020. InstaCart. Webpage.

[38] Jason T Jacques and Per Ola Kristensson. 2019. Crowdworker economics in the Gig Economy. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.

[39] Haiming Jin, Lu Su, Danyang Chen, Hongpeng Guo, Klara Nahrstedt, and Jinhui Xu. 2018. Thanos: Incentive mechanism with quality awareness for mobile crowd sensing. *IEEE Transactions on Mobile Computing* (2018).

[40] Seungwoog Jung, Seung-Ik Lee, Su-Young Chi, and Hoon Choi. 2015. POI Estimation Method with Incremental Update based on Smartphones. In *Proceedings of the 2015 International Conference on Big Data Applications and Services*. ACM, 104–110.

[41] Lei Kang and Suman Banerjee. 2017. Practical driving analytics with smartphone sensors. In *2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, 303–310.

[42] Panagiota Katsikouli, Rik Sarkar, and Jie Gao. 2014. Persistence based online signal and trajectory simplification for mobile devices. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 371–380.

[43] Jahyoung Koo and Hojung Cha. 2011. Autonomous construction of a WiFi access point map using multidimensional scaling. In *International Conference on Pervasive Computing*. Springer, 115–132.

[44] Dheeraj Kota, Neha Laumas, Urmila Shinde, Saurabh Sonalkar, Karthik Dantu, Sameera Poduri, and Gaurav S Sukhatme. 2009. deSCribe: A Personalized Tour Guide and Navigational Assistant. In *International Conference on Mobile Computing, Applications, and Services*. Springer, 393–403.

[45] Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, and Jidong Shao. 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *ninth international AAAI conference on web and social Media*.

[46] 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.

[47] Chun-Shuo Lin, Meng-Fen Chiang, Wen-Chih Peng, and Chien-Cheng Chen. 2011. An event-based POI service from microblogs. In *2011 13th Asia-Pacific Network Operations and Management Symposium*. IEEE, 1–4.

[48] Xuan Liu, Quan Yang, Shigeng Zhang, and Bin Xiao. 2020. RLLL: Accurate Relative Localization of RFID Tags with Low Latency. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[49] Zhidan Liu, Zhenjiang Li, Mo Li, Wei Xing, and Dongming Lu. 2016. Mining road network correlation for traffic estimation via compressive sensing. *IEEE Transactions on Intelligent Transportation Systems* 17, 7 (2016), 1880–1893.

[50] Jiaqing Luo and Kang G Shin. 2019. Detecting Misplaced RFID Tags on Static Shelved Items. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 378–390.

[51] Google Maps. 2019. Report an error on the map, Google Maps. https://support.google.com/maps/answer/3094088?co=GENIE.Platform%3DDesktop&hl=en. [Online; accessed 2-April-2019].

[52] Chuishi Meng, Yu Cui, Qing He, Lu Su, and Jing Gao. 2017. Travel purpose inference with GPS trajectories, POIs, and geo-tagged social media data. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 1319–1324.

[53] Chenglin Miao, Qi Li, Lu Su, Mengdi Huai, Wenjun Jiang, and Jing Gao. 2018. Attack under Disguise: An Intelligent Data Poisoning Attack Mechanism in Crowdsourcing. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 13–22.

[54] Ohio. 2019. POI Correction Officer. https://s01dasstateoh.taleo.net/careersection/oh_ext/jobdetail.ftl?job=170000KY&tz=GMT-07%3A00. [Online; accessed 2-April-2019].

[55] Avinash Parnandi, Ken Le, Pradeep Vaghela, Aalaya Kolli, Karthik Dantu, Sameera Poduri, and Gaurav S Sukhatme. 2009. Coarse in-building localization with smartphones. In *International Conference on Mobile Computing, Applications, and Services*. Springer, 343–354.

[56] Google Play. 2020. Fake GPS location. https://play.google.com/store/apps/details?id=com.lexa.fakegps&hl=en_US.

[57] 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.

[58] Sijie Ruan, Zi Xiong, Cheng Long, Yiheng Chen, Jie Bao, Tianfu He, Ruiyuan Li, Shengnan Wu, Zhongyuan Jiang, and Yu Zheng. 2020. 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. 2813–2821.

[59] Huajie Shao, Shuochao Yao, Yiran Zhao, Chao Zhang, Jinda Han, Lance Kaplan, Lu Su, and Tarek Abdelzaher. 2018. A constrained maximum likelihood estimator for unguided social sensing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2429–2437.

[60] Uber. 2020. Uber Eat. Webpage.

[61] Gang Wang, Tianyi Wang, Bolun Wang, Divya Sambasivan, Zengbin Zhang, Haitao Zheng, and Ben Y Zhao. 2015. Crowds on wall street: Extracting value from collaborative investing platforms. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 17–30.

[62] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. 2014. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 239–254.

[63] Jiangtao Wang, Feng Wang, Yasha Wang, Leye Wang, Zhaopeng Qiu, Daqing Zhang, Bin Guo, and Qin Lv. 2019. HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing. *IEEE Transactions on Mobile Computing* (2019).

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

[65] 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.

[66] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, Vol. 25. ACM, 103–114.

[67] Xiuming Zhang, Yunye Jin, Hwee-Xian Tan, and Wee-Seng Soh. 2014. CIMLoc: A crowdsourcing indoor digital map construction system for localization. In *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. IEEE, 1–6.

[68] 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), 1–25.

[69] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 1–41.

[70] 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.

[71] Meng Zhou, Ming Wang, and Qingwu Hu. 2013. A POI data update approach based on Weibo check-in data. In *2013 21st International Conference on Geoinformatics*. IEEE, 1–4.

[72] Lin Zhu, Wei Yu, Kairong Zhou, Xing Wang, Wenxing Feng, Pengyu Wang, Ning Chen, and Pei Lee. 2020. Order Fulfillment Cycle Time Estimation for On-Demand Food Delivery. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2571–2580.

[73] Xinyan Zhu and Chunhui Zhou. 2009. POI inquiries and data update based on LBS. In *2009 International Symposium on Information Engineering and Electronic Commerce*. IEEE, 730–734.