Para-Pred: Addressing Heterogeneity for City-Wide Indoor Status Estimation in On-Demand Delivery

Wei Liu*
Southeast University
220194638@seu.edu.cn

Yi Ding*
University of Minnesota
dingx447@umn.edu

Shuai Wang†
Southeast University
shuaiwang@seu.edu.cn

Yu Yang
Lehigh University
yuyang@lehigh.edu

Desheng Zhang
Rutgers University
desheng.zhang@cs.rutgers.edu

ABSTRACT

On-demand delivery is a new form of logistics where customers place orders through online platforms and the platform arranges couriers to deliver them within a short time. The acquisition of indoor status (i.e., arrival or departure at the merchants) of couriers plays an important role in order dispatching and route planning. The Bluetooth Low Energy (BLE) device is a promising solution for city-wide indoor status estimation due to the low hardware and deployment costs and low power consumption. However, the environment and smartphone model heterogeneities affect the status characteristics contained in the Bluetooth signal, resulting in the decline of status estimation performance. The previous methods to alleviate the heterogeneity are not suitable for city-wide scenarios with thousands of merchants and hundreds of smartphone models. In this paper, we propose Para-Pred, an indoor status estimation framework based on the graph neural network, which directly **Pred**icts the effective indoor status estimation model **Para**meters for unseen scenarios. Our key idea is to utilize similarity between the influence patterns of heterogeneities on the Bluetooth signal to directly infer unseen scenarios' influence patterns. We evaluate the Para-Pred on 109,378 couriers with 672 smartphone models in 12,109 merchants from an on-demand delivery company. The evaluation results show that across environment and smartphone model heterogeneities, the accuracy and recall of our method achieve 93.62% and 95.20%, outperforming state-of-the-art solutions.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Indoor Status Estimation, Wireless Sensing, Graph Neural Network

*Both authors contributed equally to this research.
†Shuai Wang is the corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539167

ACM Reference Format:
Wei Liu, Yi Ding, Shuai Wang, Yu Yang, and Desheng Zhang. 2022. Para-Pred: Addressing Heterogeneity for City-Wide Indoor Status Estimation in On-Demand Delivery. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539167

1 INTRODUCTION

On-demand delivery service is a new logistics delivery service model born in response to the Online-To-Offline Business Model (O2O). With the continuous growth of the market, many online platforms have emerged in recent years [14, 25]. In these platforms, to fulfill customers' needs, couriers deliver online orders (e.g., fresh food) from merchants (e.g., supermarkets, restaurants, etc.) to customers in a short time (e.g., 30 minutes). To make in-time delivery, it is crucial for platforms to know accurate couriers' status (i.e., arrival and departure) at merchants, which is utilized to (1) update order status in customer's APPs for better customer experience [9], (2) assign new orders to the most suitable couriers (e.g., nearby couriers) [31], and (3) train learning models to estimate the order's preparing and delivery time for future orders [35].

These are many existing works related to couriers' or humans' status detection based on techniques such as Wi-Fi [17], Bluetooth Low Energy [8, 11, 19], RFID [28], and LED lights [29]. Among all these techniques, Bluetooth Low Energy (BLE) beacons turn to be a promising solution for indoor status estimation in large-scale scenarios where physical beacons are deployed to broadcast Bluetooth signals and nearby receivers such as smartphones receive the signals. It has three key advantages. (1) *Low cost:* Each beacon costs no more than $10 [9]. (2) *Energy efficiency:* Continuous scanning in BLE only introduces less than 2% extra power consumption on couriers' smartphones [9], which is much less than Wi-Fi [6]; (3) *No modification:* No hardware modification is needed on the courier end. Because of these advantages, more and more BLE beacon systems are deployed in real-world. For example, an industrial BLE beacon system called aBeacon [9, 10] was deployed in Shanghai to infer couriers' arrival and departure status at the merchants given previously-mapped beacon-merchant pairs in the deployment, where low-cost ($ 8) advertising-only BLE beacons were deployed in 12,109 merchants.

In a controlled environment such as labs, using BLE devices to estimate arrival and departure status is straightforward. However, it has been observed that many factors in the wild impact

the performance of the BLE-based beacon system. First, a small variation in the hardware and software of smartphones may cause significant heterogeneity in the collected sensor data. That is, different smartphones respond differently even to the same status [7]. Second, the propagation path of Bluetooth signals is impacted by locations where beacon devices are deployed and surrounding environments [5]. These factors affect Bluetooth signals in different ways simultaneously, which lead to varied collected data from smartphones, even for the same smartphone and the same status. Thus, it is very challenging to use an indoor status estimation model trained with data from one scenario directly in other scenarios due to the heterogeneity.

Some approaches have been proposed to address the environment and smartphone model heterogeneities. For the smartphone model heterogeneity, conventional solutions utilize reliable but expensive sensors, which are not suitable for large-scale deployment. Furthermore, researchers apply deep neural network to learn the awareness of smartphone model heterogeneity [18, 22]. For the environment heterogeneity, recent solutions leverage transfer learning [26, 34] or adversarial learning [15, 33] to suppress the influence of environment heterogeneity on recognition tasks. These methods are difficult to be applied in city-wide scenarios with thousands of merchants and hundreds of smartphone models (e.g., 12,109 merchants and 672 smartphone models in our study), which require to re-collect a large amount of labeled data or re-train models when dealing with new scenarios.

To alleviate the environment and smartphone model heterogeneities in large-scale scenarios, we aim to design an indoor status estimation framework without new scenarios' data and model retraining. We explore the following two opportunities. (i) For smartphone model heterogeneity, couriers with different smartphones pick up orders in the same merchant, which naturally provide a scenario that a large number of heterogeneous smartphones detect signals transmitted by the same beacon. (ii) For environment heterogeneity, couriers with smartphones of a certain model pick up orders at different merchants, which results in the detection of signals transmitted from a great number of heterogeneous environments with same-model smartphones. In this way, the influence of heterogeneities is patterned and hidden in the interactive sensor data. Then our goal is to mine the influence patterns of different environments and smartphone models from the interactive data, which can be used to predict the impact of the environments and smartphone models at unseen scenarios based on the similarity between influence patterns.

Considering the opportunities, solving the large-scale indoor status estimation problem still faces the following challenges. First, there is no clear metric to quantify the influence patterns of heterogeneities on Bluetooth signals. The similarity between the influence patterns of the heterogeneities is not explicit, which has to be discovered from interactive historical sensor data, rather than being provided as ground truth knowledge. Second, how to combine the similarity to model the complex heterogeneities to predict its influence patterns on unseen scenarios is challenging. Meanwhile, the degree of similarity is different and it is not straightforward to distinguish similarities with heterogeneous strengths.

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

- To the best of our knowledge, we are the first to employ an indoor status estimation system based on the similarity information to address environment and smartphone model heterogeneities for city-wide scenarios. Based on the similarity information between influence patterns of the heterogeneities, we directly infer the effective indoor status estimation model for unseen scenarios without additional data collection and model retraining.
- To address the similarity information ambiguity, we design the similarity extraction module to capture the influence patterns of different environments and smartphone models and explore the potential similarity information. To predict the influence patterns for unseen scenarios, we design the parameter prediction module that effectively combines the similarity information to predict accurate status estimation models for unseen scenarios. Meanwhile, we design the attention mechanisms to model the heterogeneous strengths of different similarities.
- We evaluate Para-Pred on a real-world industrial BLE beacon system including 109,378 couriers with 672 smartphone models in 12,109 merchants from one of the largest O2O platforms, i.e., Eleme. Experiments show that we outperform the accuracy, recall, precision, and F1-score of the state-of-the-art methods by 11.87%, 14.54%, 10.33%, and 12.27%, respectively. In addition, we analyze the effectiveness of the similarity extraction module and the attention mechanisms. We also study the performance sensitivity of our system to different shop types and smartphone models. The results show that these designs improve the overall system performance and our model is robust for different shop types and smartphone models. We will release one month of our data for the benefits of the research community.

2 MOTIVATION AND OPPOTUNITY

2.1 Motivation

The indoor status estimation is of great significance for on-demand delivery, since couriers spend almost one-third of total working time in the indoor scene [31]. For a BLE-based indoor sensing system,Bluetooth signals received by couriers will be affected by many factors in the wild. Para-Pred addresses the problem of indoor status estimation performance degradation caused by environment and smartphone model heterogeneities. In the following, we introduce

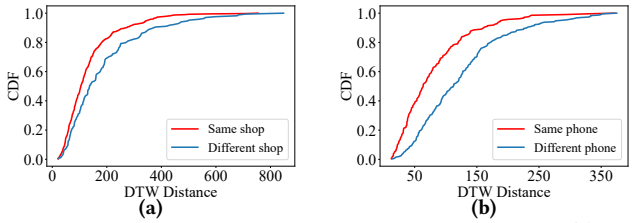

Figure 1: The Bluetooth signal patterns for different shops (a), and for different smartphones (b).

the impact of the heterogeneities on received signal strength indication (RSSI) of Bluetooth and the performance of the traditional status estimation model.

2.1.1 The impact on original Bluetooth signal data. To reflect the impact of heterogeneities on original Bluetooth signals, we utilize the dynamic time warping (DTW) distance to quantify the impact of the heterogeneities. DTW is a common algorithm to calculate the similarity between different length time series. The larger the DTW distance is, the less similar two time series are. We calculate the DTW distance between each pair of measurements collected in the same shop and different shops. The DTW cumulative distribution function (CDF) in Fig. 1a shows that the measured samples in the same shop are more similar than those measured in different shops. Similarly, Fig. 1b indicates that the samples detected by the same smartphone model are more similar than those detected by different smartphone models. They suggest that environment and smartphone models have a great impact on Bluetooth signals.

2.1.2 The impact on the status estimation model performance. To further reflect the impact of heterogeneities, we test the performance of the indoor arrival status estimation model based on CNN-LSTM [32] across different environments and different smartphone models. Data collected by phone1 in shop1 are used as the training dataset and three groups of data collected by phone1 in shop1, phone2 in shop1 and phone1 in shop2 are used as testing datasets. We use 5-fold cross-validation to split training and testing datasets, and compare the average accuracy and recall. Fig. 2 shows that for the smartphone model that is unseen in the training process, the accuracy dropped from 91.25% to 75%, and the recall dropped from 95.83% to 73.61%. For the unseen shop, the accuracy of the status estimation model drops to 71.19%, and the recall drops to 79.27%. These results emphasize the significant influence of environment and smartphone model heterogeneities on the performance of the status estimation model.

2.2 Opportunity

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

The different distributions of sensor data reflect the influence patterns of different environments and smartphone models on Bluetooth data. As shown in Fig. 3, we analyze the distribution of Bluetooth signals detected by three different smartphone models in the same shop during the same courier's pickup time. The results show

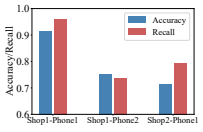

Figure 2: The performance of the traditional status estimation model.

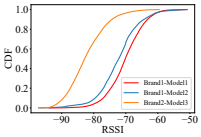

Figure 3: Similarity between the influence patterns of smartphone models.

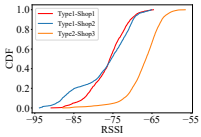

Figure 4: Similarity between the influence patterns of shops.

that data distributions of phone1 and phone2 of the same brand are more similar than that of phone3 of another brand. Similarly, we analyze the Bluetooth signal distribution of the same smartphone model in three different shops. As shown in Fig. 4, we find that the data distributions of shop1 and shop2 of the same type (i.e.milk tea) are more similar than that of another type of shop3. It shows that there are similarities between the influence patterns of different smartphone models and between the influence patterns of different environments. Furthermore, for **target domains** for which we cannot train effective status estimation models due to the lack of data, we have the opportunity to use the similarity between influence patterns of **source domains** which have sufficient data to infer the status estimation model parameters of the target domain without extra data collection.

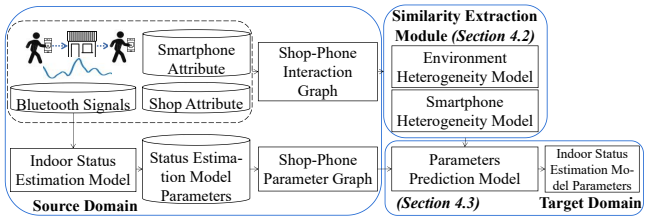

Figure 5: Framework of the Para-Pred.

3 SYSTEM & DEFINITION

3.1 System Framework

In this section, we introduce the Para-Pred framework to address the challenges mentioned in Section 1. First, we need to dig out the influence patterns of different environments and different smartphone models from a large amount of interactive sensor data, and then effectively utilize the similarity between influence patterns to predict the status estimation model parameters for the target domains, which are directly used for accurate status estimation. And we model the contributions of different similarities. Considering the above goals, we design the model framework as shown in Fig. 5, which consists of the following three modules.

Data Processing Module. This module prepares data for graph construction. First, original Bluetooth signals are processed into a suitable form as the edge attribute of the shop-phone interaction graph. Then we train the status estimation models for source domains with labeled data, and the model parameters are taken as the edge attribute of the shop-phone parameter graph. The details can be found in Section 4.1.

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

that contain the influence patterns of heterogeneities through the environment heterogeneity model and smartphone heterogeneity model. Then we construct the shop-shop similarity graph and phone-phone similarity graph according to the similarity between smartphone model latent factors and between shop latent factors. The details can be found in Section 4.2. For brevity, a smartphone in this paper represents a concrete smartphone model, e.g., Samsung Galaxy S22.

Parameters Prediction Module. We first learn effective indoor status estimation models for source domains and construct the shop-phone parameters graph based on the indoor status estimation model parameters T. Then we inherently combine T and similarity graphs G^s and G^p to predict the status estimation model parameters for the target domains. The details can be found in Section 4.3.

3.2 Problem Definition and Notations

Definition1: Indoor Status Estimation Model Parameters. We utilize the CNN-LSTM as the indoor status estimation model to estimate the probability of arrival and departure at every time clip, which consists of one-layer CNN, one-layer LSTM, and a fully connected layer. The input is the RSSI time series, and the output is the predicted probability of the arrival or departure status. For the source domain with sufficient labeled data, we obtain the status estimation model parameters by training. In section 5.2.6, we verify that our status estimation model structure is sufficient for status estimation.

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

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

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

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

4 DESIGN OF PARA-PRED

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

4.1 Data Processing Module

First, due to the packet loss in the process of data collection, we interpolate the RSSI time series and utilize the Kalman filter [23] to remove the abnormal values. To process the data into the form required by the shop-phone interaction graph G, we utilize DTW to align all RSSI time-series detected by the same smartphone model in the same shop during pickup time. Then, we calculate the mean vector $e_{ij} = \frac{1}{L}\sum_{l=1}^{L} r_l$ as the attribute of E, where r_l is the lth historical Bluetooth signal time-series detected by smartphone model v_j in shop u_i, L is the number of time series.

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

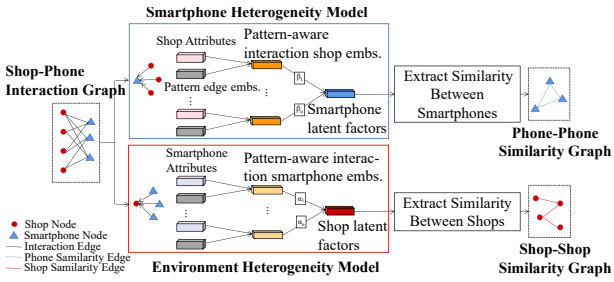

Figure 6: Structure of Similarity Extraction Module.

4.2 Similarity Extraction Module

The similarity extraction module aims to learn the influence patterns of heterogeneities from historical interaction data and obtain the similarity between the influence patterns of environments and between the influence patterns of smartphone models. We define the shop-phone interaction graph G and utilize the environment heterogeneity model and smartphone heterogeneity model to learn latent factors of shop and smartphone model nodes that contain their influence patterns and output the shop-shop similarity graph G^s and phone-phone similarity graph G^p as shown in Fig. 6. Next, we introduce the environment heterogeneity model, phone heterogeneity model, and how to obtain the similarity graphs in detail.

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

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

$$\alpha^*_{ix} = \sigma(W \cdot [\{g_{ix}, \forall x \in P(i)\} \parallel s_i] + b) \tag{1}$$

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

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

$$h_i = \sigma(W \cdot \left\{ \sum_{x\in P(i)} \alpha_{ix} g_{ix} \right\} + b) \tag{2}$$

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

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

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

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

$$L_r = \frac{1}{|Q|} \sum_{i,j\in Q} \left\| \hat{e}_{ij} - e_{ij} \right\|_2 \tag{3}$$

where |Q| is the number of observed interactive edges and e_{ij} is the ground truth edges in the shop-phone interaction graph. The latent factors of shops and smartphone models are utilized to construct similarity graphs in Section 4.2.4.

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

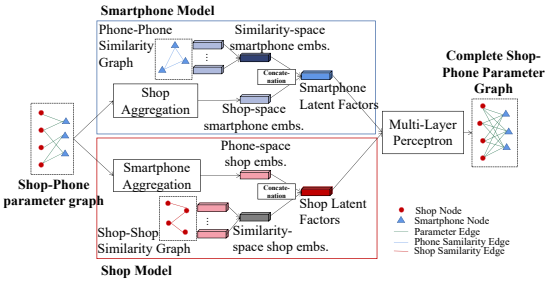

Figure 7: Structure of Parameters Prediction Module.

4.3 Parameters Prediction Module

The parameters prediction module infers the parameters of indoor status estimation models for target domains based on (1) the similarity between shops'influence patterns and (2) the similarity between smartphone models'influence patterns, respectively. As shown in Fig. 7, the input of this module is a shop-phone heterogeneous graph composed of three sub-graphs: i.e., the shop-shop similarity graph G^s, phone-phone similarity graph G^p and incomplete shop-phone parameter graph T. The output is a complete shop-phone parameter graph, where we obtain the effective indoor status estimation model for the target domain according to the predicted status estimation model parameters.

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

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

$$h_i^{t1} = \sigma(W \cdot \left\{ \sum_{a \in P(i)} \gamma_{ia} q_{ia} \right\} + b) \tag{4}$$

where q_{ia} is the parameter-aware smartphone model embedding which models the interaction and model parameter information between shop u_i and smartphone model v_a. To obtain the q_{ia}, we first introduce a parameter embedding n_{ia} to model parameters between u_i and v_a by inputting the edge attribute e_{ia}^t to a MLP φ_p, i.e., $n_{ia} = \varphi_p(e_{ia}^t)$. Then we feed the concatenation of n_{ia} and the smartphone model v_a's attribute p_a into a MLP φ_v. And q_{ia} is defined as $q_{ia} = \varphi_v([n_{ia}||p_a])$. We design a **smartphone attention** γ_{ia} to represent the importance of the interaction with smartphone model v_a in contributing to phone-space shop embedding h_i^{t1}. We input the parameter-aware smartphone model embedding q_{ia} and shop u_i's attribute s_i into a one-layer neural network to obtain the smartphone attention γ_{ia}, i.e., $\gamma_{ia} = \frac{exp(\sigma(W \cdot [q_{ia}||s_i]+b))}{\sum_{a \in P(i)} exp(\sigma(W \cdot [q_{ia}||s_i]+b))}$.

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

$$h_i^{t2} = \sigma(W \cdot \left\{ \sum_{c \in C(i)} \mu_{ic} h_c^{t1} \right\} + b) \tag{5}$$

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

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

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

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

$$\hat{e}_{ij}^{t(k)} = \sigma(W^{(k)} \cdot e_{ij}^{t(k-1)} + b^{(k)}) \tag{6}$$

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

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

$$L_p = \frac{1}{|O|} \sum_{i,j \in O} \left\| \hat{e}_{ij}^t - e_{ij}^t \right\|_2 \tag{7}$$

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

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

$$L_s = \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} -y_{im} log y'_{im} \tag{8}$$

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

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

4.4 Model Training

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

5 EVALUATION

5.1 Experiment Setup

5.1.1 Data Collected.

aBeacon Monitoring Data: The aBeacon monitoring dataset includes 64 million delivery orders involved with 109,378 couriers with 672 smartphone models in 12,109 merchants in Shanghai. The period of this dataset is from 2019.07.01 to 2020.10.31. Due to the agreement with the on-demand delivery services company, we are able to show the evaluation results in a sub-region of Shanghai (i.e., 6km × 6km) including 269 active couriers with 143 smartphone models, 187 active merchants, and around 4341 delivery orders on each day in four months.

Courier Report Data: For each delivery order, the report data record the time of four major status, i.e., accepting an order, arrival at the merchant, departure from the merchant (with the order), and final delivery to the customer. We obtain the status of couriers because these timestamps are uploaded by courier smartphone apps. The primary attribute, detailed information of the dataset and the explanation of ground truth are shown in Appendix A.2.

5.1.2 Parameter Settings. We implement our method and baselines with Pytorch 1.3.1 in Python 3.7.5 environment and train these with 16 GB memory and Tesla V100-SXM2 GPU. The four-month aBeacon monitoring data are divided into four months, we utilize the first three months' data to train our model and the last month's data as the evaluation. How to determine various parameters in our model is discussed in Appendix A.3.

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

- SVM [24]: This model utilizes the support vector machine as the classifier, we extract the time-domain features including mean, standard deviation, skewness, kurtosis, max, min, shape factor and the frequency-domain features including FFT Peaks, energy, and domain-frequency.
- C-LSTM [32]: This method discovers deep features from time series for status estimation without extracting features manually.
- CrossSense [34]: This method proposes a roaming model based on machine learning to transform the signal features from source domains to target domains and adopts multiple expert models for recognition.
- EUI [33]: This method contains two domain discriminators to classify the user or the position of the gesture with a multi-task optimization in the environment and user invariant training via adversarial learning. In our problem, we convert to design the environment and smartphone model domain discriminators.
- GraphSage [13]: This is a graph method, which transfers information through the edge without using the attributes of nodes and edges to attend neighbor nodes. We use this method to learn the shop-phone parameter graph for the status estimation model parameter prediction.

5.2 Result Analysis

5.2.1 Overall performance. We compare our work with the above baselines, and the average results of accuracy, recall, precision, and F1-score of arrival and departure status are shown in Table 1. We observe that our method has the best overall performance compared with other methods. In addition, our method does not need extra data re-collection and model retraining when facing new target domains. Para-Pred achieves more than 17% higher precision and F1-score than the classic baselines(i.e., SVM and C-LSTM), because we consider the impact of the heterogeneities. The average accuracy and recall of Para-Pred are 93.62%, and 95.20%, respectively, which advantage the EUI by 11.87% and 14.54%. Then, the advantage of our Para-Pred over GraphSage validates the effectiveness of our aggregation method and attention mechanisms. We also conduct a t-test to show that our results are statistically significant with the p-value < 0.01 compared to the best baseline EUI.

Table 1: Overall performance

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

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

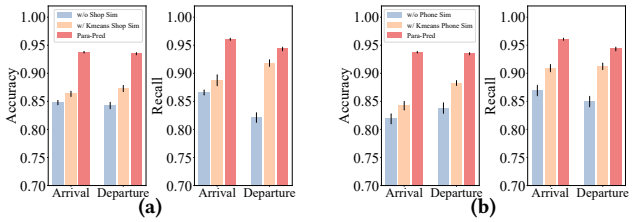

Figure 8: Significance of Similarity Extraction Module.

5.2.2 Significance of Similarity extraction module. This section analyzes the importance of the similarity learned by the similarity extraction module for system performance.

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

The comparison results are shown in Fig. 8a, the average accuracy and recall of arrival and departure status of our method are higher than 93.0% and 94.3% severally, which is better than these two variants. The w/o Shop Sim method performing worst proves the importance of the similarity information between influence patterns of shops. Our Para-Pred performing better than the w/ Kmeans Shop Sim method proves the advantage of our environment heterogeneity model which learns influence patterns of environments and obtains the effective similarity information by considering the interaction information contained in sensor data.

The analysis of w/ and w/o the similarity between smartphone influence patterns. Similarly, to intuitively show how the similarity between smartphone influence patterns learned by

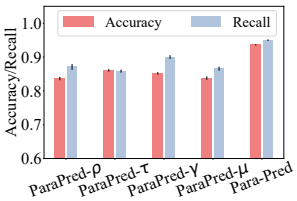

Figure 9: The importance of attention mechanisms.

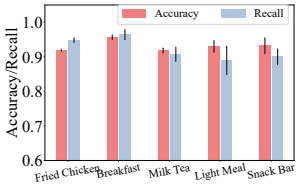

Figure 10: The performance of different environments.

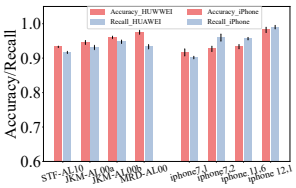

Figure 11: The performance of different smartphone models.

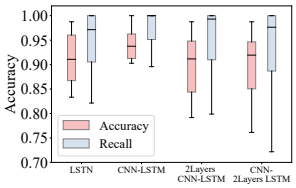

Figure 12: Results for Different Model Structures.

the smartphone model heterogeneity model affects our model performance, we evaluate our model and two variants, **w/o Phone Sim** and **w/ Kmeans Phone Sim** which are defined in Appendix A.4. Fig. 8b shows that our Para-Pred performs best, it proves that the smartphone heterogeneity model extracts effective smartphone model influence patterns and the similarity information is important to improve the performance of final status estimation.

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

5.2.3 The importance of attention mechanisms. In the parameter prediction module, we design the attention mechanisms based on the edge and node attributes to aggregate features of neighbors. This module involves four different attention mechanisms, including shop attention ρ, smartphone similar neighbor attention τ, smartphone attention γ, and shop similar neighbor attention μ. To study the impact of different attention mechanisms on system performance, we compare the performance of four variants (i.e., ParaPred-ρ, ParaPred-τ, ParaPred-γ and ParaPred-μ) which all use mean aggregation instead of aggregation based on our four attention mechanisms, respectively.

The results in Fig. 9 show that removing any attention mechanism leads to the final performance degradation. The attention mechanisms of our model efficiently distinguish the different contributions of different neighbors and improve the system performance.

5.2.4 Impact of Environment Diversity. The purpose of this section is to study the impact of different environment types on the performance of Para-Pred. Taking the arrival estimation as an example, we select the samples detected by the same smartphone in five different types (fried chicken, breakfast, milk tea, light meal, and snack bar) of shops for testing. These five types are common in daily life and have different environmental characteristics. As shown in Fig. 10, the performance of different shop types is slightly different, the average accuracy is more than 91% and the average recall is more than 89% in all types of shops. In general, our framework is robust for different environment types.

5.2.5 Impact of Smartphone Diversity. To verify the performance of Para-Pred in different smartphone models, we test our model with the data detected by eight different smartphone models under two well-known brands (HUAWEI and iPhone) in the same shop for arrival estimation. The abscissa in Fig. 11 is sorted by the release time of smartphones from the same brand, and the results show that iPhone 12,1 has the highest average accuracy and recall of 98.20% and 99.09%, respectively. We find for smartphone models from the same brand, the newer the smartphone, the better the average performance. In a nutshell, our model performs well for different smartphone models.

5.2.6 Comparison of Different Status Estimation Model Structures. As shown in Fig. 12, compared with the simple LSTM model, the average accuracy and recall of CNN-LSTM structure are improved by around 5% and 7%, respectively. Because the convolution layer can extract high-level spatial features from raw Bluetooth signals. Moreover, we analyze the effect of the number of neural network layers on our model performance. We test two convolutional layers CNN-LSTM model and two LSTM layers CNN-LSTM model. The results show that increasing the number of layers does not further improve the performance. Therefore, considering the trade-off between the training time and performance, we choose CNN-LSTM as our indoor status estimation model.

6 DISCUSSION

In designing and building the system, we have some lessons learned on the effectiveness of graph learning in solving the sensing heterogeneity problem and the generalizability of the solution. A detailed discussion can be found in Appebdix B.

7 RELATED WORK

7.1 Cross-domain Learning Methods

The research on how to improve the cross-domain generalization ability of the recognition framework fall into two categories: one is the method based on data conversion [18, 26, 34], and the other is the method based on adversarial learning [15, 33]. The main idea of the former is to learn how to transform the data features collected from one domain into the data features of other domains for the recognition model training. For example, CrossSense [34] learns a roaming model that transforms data features for each target domain and uses the generated data features for downstream tasks training.

The latter type uses the idea of adversarial learning to extract features irrelevant to the environment or user for recognition. For example, RFID [33] designs two domain discriminators to classify the user who performs this gesture or positions where the gesture is executed to extracts features irrelevant to the environments and users. However, when a new target domain appears, all these methods require re-collecting new data under limited conditions and model retraining, which are not efficient for large-scale scenarios. Our method infers the status estimation model for the new target domain according to the similarity between heterogeneities learned by the similarity extraction module without additional data collection and model retraining.

7.2 Graph Embedding based on GNN

The purpose of graph embedding is to map graph data into low dimensional dense vectors. It captures the topology of the graph, the relationship between vertices, and other related information. The graph neural network (GNN) model first appeared in [20], it extends

the existing traditional neural network model and can be used to process data with arbitrary graph structure. Since GNNS effectively learn on graph structure data, there are several advanced graph embedding methods based on GNN. GraphSage [13] learns the nodes embeddings by aggregating information from neighbors in an inductive manner. Graph autoencoder (GAE) and variational graph autoencoder (VGAE) [16] use graph convolution network (GCN) encoder and simple inner product decoder to learn node embedding by minimizing the reconstruction error of adjacency matrix while considering graph structure and node feature information. [27] [12] further introduce the attention mechanism to learn the importance weights of neighbor nodes and the edge content. In our method, we incorporate the node and edge attributes and utilize convolutional network and attention mechanisms to fully explore the features.

7.3 RSSI-based Sensing

Received Signal Strength Indication (RSSI) can be used to identify human activities [4, 21]. For example, in-air hand gestures around the user's mobile device can be recognized by analyzing the change of WiFi signal strength [4]. In recent years, the number of smartphones and devices equipped with BLE functions has increased. Therefore, many BLE sensing systems have been proposed. For example, [30] applies the idea of global map matching to route estimation based on BLE beacons. Given the known mapping of the BLE beacons and the couriers' smartphones, a straightforward solution is to mark the courier as 'arrival' when (s)he first scans the beacon. However, it is inaccurate and a detailed explanation can be found in Appendix A.5.

8 CONCLUSION

In this work, we propose Para-Pred, an city-wide indoor status estimation system based on the similarity information to address environment and smartphone heterogeneities with thousands of merchants and hundreds of smartphone models. We design the similarity extraction module to explore the similarity between influence patterns of heterogeneities, xian zxi an zaand the parameters prediction module to infer the effective status estimation model for the new target domain based on the learned similarity information. We evaluate our method on a real-world dataset from an on-demand delivery company. The experiment results show that the average status estimation accuracy and recall of our method outperforms other state-of-the-art methods by 11.87% and 14.54%, respectively.

ACKNOWLEDGMENTS

This work was supported in part by Science and Technology Innovation 2030 - Major Project 2021ZD0114202.

REFERENCES

[1] 2016. Airport Benchmarking. Airlines and airports are beaconizing. https://www.airportbenchmarking.com/airlines-and-airports-are-beaconizing/.
[2] 2020. Government of Singapore. Help speed up contact tracing with tracetogether. https://www.gov.sg/article/help-speed-up-contact-tracing-with-tracetogether/.
[3] 2020. Locatify. Eldheimar museum, intuitively designed beacon based museum audio guide. https://locatify.com/blog/case-studies/eldheimar-museum/.
[4] Heba Abdelnasser, Moustafa Youssef, and Khaled A. Harras. 2015. WiGest: A ubiquitous WiFi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 1472–1480.
[5] H. S. Ahn and W. Yu. 2009. Environmental-Adaptive RSSI-Based Indoor Localization. *IEEE Transactions on Automation Science & Engineering* 6, 4 (2009).
[6] Ganesh Ananthanarayanan and Ion Stoica. 2009. Blue-Fi: Enhancing Wi-Fi Performance Using Bluetooth Signals *(MobiSys '09)*. 249–262.
[7] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. 2014. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable. In *Network and Distributed System Security Symposium*.
[8] Yi Ding, Dongzhe Jiang, Yu Yang, Yunhuai Liu, Tian He, and Desheng Zhang. 2022. P2-Loc: A Person-2-Person Indoor Localization System in On-Demand Delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–24.
[9] 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 *USENIX NSDI*. 859–872.
[10] Yi Ding, Yu Yang, Wenchao Jiang, Yunhuai Liu, Tian He, and Desheng Zhang. 2021. Nationwide deployment and operation of a virtual arrival detection system in the wild. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 705–717.
[11] Cole Gleason, Dragan Ahmetovic, Saiph Savage, Carlos Toxtli, and Jeffrey P. Bigham. 2018. Crowdsourcing the Installation and Maintenance of Indoor Localization Infrastructure to Support Blind Navigation. *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–25.
[12] Liyu Gong and Qiang Cheng. 2019. Exploiting Edge Features for Graph Neural Networks. In *CVPR*. 9211–9219.
[13] W. L. Hamilton, R. Ying, and J. Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.
[14] InstaCart. 2021. InstaCart. https://www.instacart.comWebpage.
[15] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, and Yaqing Wang. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *MobiCom*. 289–304.
[16] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. arXiv:1611.07308 [stat.ML]
[17] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. 2014. Accurate indoor localization with zero start-up cost. In *MobiCom*. 483–494.
[18] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Velickovic, and Pietro Lio. 2018. Using Deep Data Augmentation Training to Address Software and Hardware Heterogeneities in Wearable and Smartphone Sensing Devices. In *IPSN*.
[19] Rachael Purta and Aaron Striegel. 2017. Estimating dining hall usage using bluetooth low energy beacons. In *UbiComp/ISWC*. 518–523.
[20] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* (2009), 61–80.
[21] Markus Scholz, Till Riedel, Mario Hock, and Michael Beigl. 2013. Device-free and device-bound activity recognition using radio signal strength. In *Augmented Human International Conference*. 100–107.
[22] S. S. Shubha, T. Sen, H. Shen, and M. Normansell. 2021. A Diverse Noise-Resilient DNN Ensemble Model on Edge Devices for Time-Series Data. In *SECON*. 1–9.
[23] Sam Shue and James M. Conrad. 2016. Reducing the Effect of Signal Multipath Fading in RSSI-Distance Estimation Using Kalman Filters. In *SpringSim (CNS)*. 5.
[24] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, and Mads Mller Jensen. 2015. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *SenSys*. 127–140.
[25] Uber. 2021. Uber Eat. https://www.ubereats.comWebpage.
[26] Aditya Virmani and Muhammad Shahzad. 2017. Position and Orientation Agnostic Gesture Recognition Using WiFi. In *MobiSys*. 252–264.
[27] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed Graph Clustering: A Deep Attentional Embedding Approach. In *IJCAI*. 3670–3676.
[28] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. 2019. Are RFID Sensing Systems Ready for the Real World?. In *MobiSys*. 366–377.
[29] Yu Lin Wei, Chang Jung Huang, Hsin Mu Tsai, and Ching Ju Lin. 2017. CELLI: Indoor Positioning using Polarized Sweeping Light Beams. In *MobiSys*. 136–147.
[30] Daisuke Yamamoto, Ryosuke Tanaka, Shinsuke Kajioka, Hiroshi Matsuo, and Naohisa Takahashi. 2018. Global map matching using BLE beacons for indoor route and stay estimation. In *SIGSPATIAL/GIS*. 309–318.
[31] 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 *MobiCom*. 1–14.
[32] S. Yu, Q. Long, and Q. Yin. 2018. A C-LSTM Neural Network for Human Activity Recognition Using Wearables. In *ISSI*.
[33] Yinggang Yu, Dong Wang, Run Zhao, and Qian Zhang. 2019. RFID Based Real-Time Recognition of Ongoing Gesture with Adversarial Learning. In *SenSys*.
[34] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. 2018. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In *MobiCom*. 305–320.
[35] L. Zhu, W. Yu, K. Zhou, X. Wang, and L. Pei. 2020. Order Fulfillment Cycle Time Estimation for On-Demand Food Delivery. In *SIGKDD*. 2571–2580.

A APPENDIX

A.1 Mathematical Notations

The mathematical notations used in this paper are listed in Table 2.

A.2 Data Collect

A.2.1 Data Interpretation, Privacy, and Data Release. We visualize the actual deployment of aBeaconas Fig. 13. Every time an aBeacon device broadcast is received by a courier's smartphone, we record the information of the aBeacon device, smartphone, and the Received Signal Strength Index (RSSI) of the broadcast. The primary attribute and detailed information of the aBeacon Monitoring Data and the Courier Report Data are shown in Table 3 and Table 4, respectively. The aBeacon monitoring data and courier report data utilized in our work are collected under the consent agreement of couriers by our platforms. All data are anonymized, and any ID information cannot be tracked or identified in practice. Moreover, our data does not involve the couriers' personal information, e.g., age, gender, income, to protect the privacy of couriers. We will release one month of our data collected in aBeacon platform for the research community to validate our results and conduct further research.

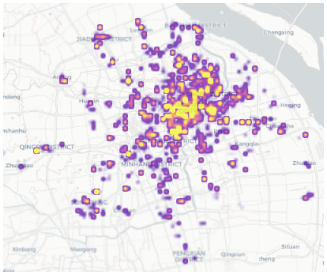

Figure 13: Beacon device heatmaps in Shanghai.

A.2.2 Ground Truth. We follow the idea in [31] that (1) most couriers' outdoor/indoor reporting behaviors are intrinsically consistent under certain context, i.e., if a courier has accurate reporting behavior at outside-merchants, (s)he generally has accurate reporting behavior at inside-merchants, and (2) the distance between the courier and merchant obtained from the historical GPS trajectories follows a common trend during pickup time, i.e., changes from decreasing to increasing, where the shortest distance in the trend corresponds to the arrival time. In this way, we define a order as a **reliable orders** if the absolute difference between the arrival time reported by the courier at outside-merchant and the arrival time obtained by the GPS trajectory is less than 1s. And the courier whose reliable orders account for more than 90% of his total orders is defined as the **reliable courier**. We collect the arrival (departure) time reported by reliable couriers as the ground truth.

Table 3: aBeacon Monitoring Data

Attribute	Example
Courier ID	C_000001
Timestamp	2019/07/15 12:30:23
Device ID Tuple	(UUID, Major, Minor)
Merchant ID	M_000001
RSSI	-70dB
Phone ID	D_000001
Phone Brand/OS	Apple/iOS
Phone Model	iPhone X

Table 4: Courier Report Data

Attribute	Example
Courier ID	C_000001
Timestamp	2019/07/15 12:30:23
Merchant ID	M_000001
Order ID	O_000001
Report Type	Acceptance/Arrival/ Departure/Delivery

A.3 Parameter Selection

For the shop-phone interaction graph and parameter graph, we randomly split the edges into 5 folds for cross-validation in all the reported experiments involving graph learning. In the status estimation model, the CNN layer contains a convolution kernel of size 4, stride 1, and ReLU activation function. The LSTM layer contains 12 hidden cells, and the fully connected layer has 8 hidden cells. In the similarity extraction module and parameter prediction

Table 2: Mathematical Notations

Notations	Definitions	Notations	Definitions
r_l	The lth time series of RSSI received by smartphone v_j from shop u_i	e_{ij}	The interactive edge attribute of smartphone v_j in shop u_i
s_i	The shop u_i's attribute	p_j	The smartphone v_j's attribute
$P(i)$	The set of smartphones which shop u_i interacted with	$S(j)$	The set of shops which smartphone v_j interacted with
h_i	The latent factor of shop u_i	z_j	The latent factor of smartphone v_j
α_{ix}	The smartphone attention of smartphone v_x in contribution to h_i	β_{jo}	The shop attention of shop u_o in contribution to z_j
$e^t_{i,j}$	The parameter edge attribute between smartphone v_j and shop u_i	$C(i)$	The set of shops which shop u_i similar to
$N(j)$	The set of smartphones which smartphone v_j similar to	h^t_i	The latent factor of shop u_i of shop-phone parameter graph
z^t_j	The latent factor of smartphone v_j of shop-phone parameter graph	$\hat{e}^t_{ij}$	The predicted status estimation model parameters for smartphone v_j in shop u_i

module, the embedding dimension of all graphs is 32, the similarity threshold ϵ is 0.2, and the trade-off parameter η is 1. We check how these two parameters affect Para-Pred performance in Appendix A.2.1. We use Adam optimizer in the training process, the learning rate is 0.0001. The activation function of the network layer is ReLU. To prevent overfitting, we add one BatchNorm layer after the fully connected layer in our graph model.

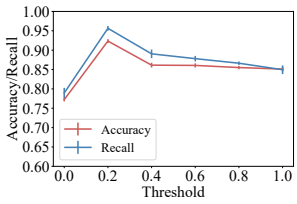

Figure 14: Parameter sensitivity of different thresholds.

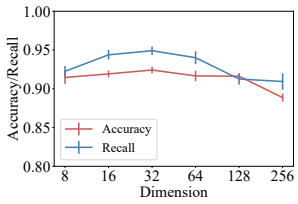

Figure 15: Impact on embedding dimension.

A.3.1 Parameter Study. This section studies the influence of distance threshold ϵ and the embedding dimensions on the system performance. The distance threshold ϵ determines the sparsity of similarity graphs. As shown in Fig. 14, we analyze the change of system performance when the distance threshold ϵ increases from 0 to 1. The best distance threshold is 0.2.

Then we analyze the impact of the embedding dimension on the system performance. Fig. 15 shows the performance comparison. Generally, with the increase of embedding dimension, the performance first increases and then decreases. When the embedding dimension increases from 8 to 32, the performance is significantly improved. And when the embedding dimension is 64, the performance of the model decreases. It shows that the larger the embedding dimension is, the stronger the ability of information expression is. When the dimension is too great, the vector will be sparse, and the complexity of the model will be improved. Considering the trade-off between accuracy and computational complexity, we select 32 as the default setting of the embedding dimension in our algorithm.

A.4 The definition of w/o Phone Sim and w/ Kmeans Phone Sim

(1) w/o Phone Sim: We remove the phone-phone similarity graph from the parameter prediction module and the smartphone latent factors are learned only from the shop-phone parameter graph. **(2) w/ Kmeans Phone Sim:** We cluster the raw sensor data based on Kmeans, and connect smartphones in the same class to construct a new phone-phone similarity graph instead of learning from the similarity extraction module.

A.5 Why straightforward solutions for status estimation do not work.

Given the known mapping of the BLE beacons and the couriers' smartphones, a straightforward solution is to mark the courier as 'arrival' when (s)he first scans the beacon. However, it is inaccurate because it will lead to false positive (e.g., passing by but detected as arrival) and false negative (e.g., arrival but detected late due to improper deployment). For example, if two merchants are located close to each other, a courier may receive ID tuples from multiple merchants' beacons at the same time, and when the beacon is deployed too close to the door, the ID tuple is received even if the courier does not arrive at the merchant. Therefore, we should make full use of the information of the Bluetooth signal strength received by the couriers over time to detect arrival and departure status accurately. The existing BLE-based studies do not consider the impact of heterogeneities on recognition performance in large-scale deployment. Our work effectively alleviates the influence of heterogeneities and improves the indoor status estimation performance.

B DETAILED DISCUSSION

B.1 Lesson Learned

(1) We analyze the influence patterns of the environment and smartphone heterogeneities and find that there is similarity information in them as shown in Fig. 3 and Fig. 4. (2) The similarity information is important for predicting effective indoor status estimation models for target domains (i.e., Section 5.2.2). (3) Graph learning works well for addressing the heterogeneities in large-scale indoor status estimation since the influence patterns of heterogeneities and the similarity information can be effectively learned and represented. (4) The evaluation results show that our method alleviates the heterogeneities effectively for indoor status estimation in large-scale scenarios (Table 1). We also find that our method is robust for different types of environments (Fig. 10), and the release time of smartphone models from the same brand has a positive impact on system performance (Fig. 11).

B.2 Generalizability

Although Para-Pred is designed for on-demand delivery, we believe the underlying ideas of addressing the impact of heterogeneities on sensing data can be generalized to other BLE scenarios, such as interaction in museums[3], airports[1], and TraceTogether[2]. For example, in airports[1], passengers receive tailored information as they arrive at the lounge or retail areas. In this scenario, Bluetooth signals detected by different smartphones are input to the similarity extraction module to learn the similarity between the effects of heterogeneities. The parameter prediction module predicts the status estimation model parameters for new scenarios. We believe Para-Pred with some modifications is a potential solution to mitigate the heterogeneity effects in these application scenarios, which provides more accurate status estimation and better services to users.