CoMiner: Nationwide Behavior-driven Unsupervised Spatial Coordinate Mining from Uncertain Delivery Events

Zhiqing Hong
Rutgers University, Piscataway, USA
JD Logistics, Beijing, China
zhiqing.hong@rutgers.edu

Guang Wang
Florida State University
Tallahassee, USA
guang@cs.fsu.edu

Wenjun Lyu
Rutgers University
Piscataway, USA
wenjun.lyu@rutgers.edu

Baoshen Guo
Southeast University
Nanjing, China
guobaoshen@seu.edu.cn

Yi Ding
University of Minnesota
Minneapolis, USA
dingx447@umn.edu

Haotian Wang
JD Logistics
Beijing, China
wanghaotian18@jd.com

Shuai Wang
Southeast University
Nanjing, China
shuaiwang@seu.edu.cn

Yunhuai Liu
Peking University
Beijing, China
yunhuai.liu@pku.edu.cn

Desheng Zhang
Rutgers University
Piscataway, USA
desheng@cs.rutgers.edu

ABSTRACT

Geocoding, associating textual addresses with corresponding GPS coordinates, is vital for many location-based services (e.g., logistics, ridesharing, and social networks). One of the most common Geocoding solutions is using commercial map services (e.g., Google Maps) by uploading textual addresses to obtain corresponding coordinates. However, this is typically not practical for some location-based service providers due to real-world challenges like commercial competition and high costs (recurring fees). In this paper, we design a new cost-effective Geocoding framework to automatically infer the geographic coordinates from textual addresses for service providers. To achieve this, we take the E-Commerce logistics service as a concrete scenario and design `CoMiner`, an unsupervised coordinate inference framework based on textual address data, delivery event data, and courier trajectory data. There are three main components in `CoMiner`. (1) A POI-level clustering model by modeling customers' shopping patterns at different spatial granularities; (2) A Delivery Mobility Graph (*DMG*) by modeling couriers' delivery events and geographic coordinates; (3) A behavior-driven address ranking model by mining couriers' uncertain reporting behaviors to further infer coordinates on *DMG*. We extensively verify the performance of `CoMiner` with a three-phase evaluation from data-driven experiments to real-world deployment. (i) We conduct extensive experiments on three large-scale datasets where `CoMiner` achieves an average accuracy of 95.1%, which outperforms the state-of-the-art methods by 20.3%. (ii) We deploy `CoMiner` in JD Logistics, inferring coordinates for over 30 million addresses with an average accuracy of 93.3%. (iii) We utilize `CoMiner` for two Geocoding-based applications, i.e., parcel re-routing optimization and abnormal delivery event detection.

CCS CONCEPTS

• **Information systems** → **Location based services**; • **Human-centered computing** → **Ubiquitous and mobile computing**.

KEYWORDS

Spatio-temporal data mining, coordinate inference, human behavior

ACM Reference Format:
Zhiqing Hong, Guang Wang, Wenjun Lyu, Baoshen Guo, Yi Ding, Haotian Wang, Shuai Wang, Yunhuai Liu, and Desheng Zhang. 2022. CoMiner: Nationwide Behavior-driven Unsupervised Spatial Coordinate Mining from Uncertain Delivery Events. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3557915.3560944

1 INTRODUCTION

In recent years, location-based services have received growing interests from both industry and academia, e.g., logistics [1, 3], E-Commerce (e.g., Amazon, JD.COM), and on-demand services (e.g., UberEats and Meituan [32]). Unlike online services such as item recommendation, location-based services consider location as an essential factor in service designs, e.g., user-entered addresses are needed for E-commerce parcel delivery. Therefore, Geocoding, i.e., accurately matching users' textual addresses with the corresponding geographic coordinates or inferring geographic coordinates (coordinates thereafter) of textual addresses, is essential. Taking E-commerce logistics as a concrete example, users place an order on an E-Commerce platform, and the order parcel will be delivered to the user-entered address within a given time. In this delivery process, inferring accurate coordinates for users' addresses is very important, directly determining the delivery service's efficiency. Inaccurate coordinates will result in dispatching a parcel to the wrong delivery station (a city is partitioned into many delivery stations for parcel delivery), which results in an additional parcel transfer

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11…$15.00
https://doi.org/10.1145/3557915.3560944

from the wrong station to the right station and then delivering it to users. This additional transfer process is called *re-routing*, which causes significant environmental and economic waste.

In industry, a common method to infer coordinates is utilizing commercial maps' Geocoding (GC) services, e.g., Google Maps and Baidu Maps. Even with their convenience, commercial GC services pose two challenges for logistics companies. **(i) Low Accuracy.** In E-Commerce logistics, text addresses entered by users have some irregular and complex formats, due to reasons such as weak regulation on address formats and urban development. Especially in developing countries such as China and India, it is normal to have unstructured, partially missing or ambiguous addresses [33, 35], which greatly decreases the accuracy of commercial GC services. Moreover, commercial GC services are further challenged in rural areas because of the address data collection hardness and data sparsity. We further systematically evaluate the GC performance of Gaode, Baidu, and Tencent Maps services, and the average 300-meter GC accuracy is 80.9% nationwide, including rural areas, which is similar to some published GC accuracy results in [34, 35]. **(ii) Considerable Cost.** Although a map app user can use commercial GC services for free, an E-Commerce platform needs to pay by GC service request frequency. It is a considerable cost for these platforms with millions of orders to be Geocoded. Another industrial approach is building <address, coordinates> dictionary and growing this dictionary by couriers' reporting coordinates for new addresses. However, the main limitation is that couriers' reporting coordinates have significant uncertainty, i.e., the reporting coordinates are sometimes inaccurate (see Sec. 2.3).

In academia, the drawbacks of commercial GC motivate research communities to explore other solutions, e.g., crowdsourcing [18, 26] and utilizing open-source datasets [29, 39]. In some research works, logistics data is used for for GC-related applications [31, 33]. However, these studies either assume the delivery data is of good quality (e.g., a high sampling rate of trajectory data) [31] or reporting behaviors are stable [33]. These assumptions are hard to hold in practice in a real-world scenario with large-scale low-quality data and unreliable reporting behaviors, which may lead to unsatisfactory performance (see Sec. 4.2).

Recently, the ubiquitous usage of GPS-enabled devices in E-Commerce logistics has brought a great opportunity for solving the coordinate inference problem. The delivery couriers are familiar with a specific spatial area based on experiences, which ensures the success of parcel delivery. Most of the time, they directly navigate to the input textual addresses according to their experiences. Moreover, due to the commercial insurance requirement and the quality of satisfaction improvement, couriers' GPS trajectories are recorded and uploaded in real time. They also need to report delivery events (by confirming on their Personal Digital Assistant (PDA)) when finishing delivery at users' locations. These couriers' behaviors provide a great opportunity for us to infer GPS coordinates. However, inferring the coordinates for addresses from delivery trajectories is nontrivial due to couriers' delivery behavior variance. For example, delivery events may be reported before the actual delivery time due to delivery deadlines or heavy delivery tasks. (see Sec. 2.3)

To tackle the challenges mentioned above, we design `CoMiner`, an unsupervised spatial coordinate mining framework, to infer the GPS coordinates for given text addresses with low costs and high accuracy. After an in-depth data-driven investigation of address and POI data, `CoMiner` first infers coordinates for some addresses by a POI-level clustering model. Then, to infer coordinates that cannot be inferred by the clustering model, we design a Mobility Graph Construction module that formulates all addresses with both known and unknown coordinates on a graph. Based on this graph, we design a behavior-driven ranking module to infer coordinates for unknown addresses (coordinates are unknown) by modeling couriers' reporting behavior and designing a stay-point ranking algorithm. In summary, this paper makes four contributions:

- To the best of our knowledge, `CoMiner` is the first nationwide, behavior-driven coordinate inference framework for textual addresses. Specifically, we implement `CoMiner` based on E-commerce logistics delivery data, i.e., delivery trajectories, delivery events, and waybills. The design insight of `CoMiner` is based on large-scale datasets with more than 120,000 professional couriers in over 600 cities from an E-Commerce logistics platform in China.
- To address the challenge of couriers' reporting behavior uncertainty, we design an unsupervised coordinate mining model. Instead of clustering coordinates of addresses, we cluster at the POI level by mining POI entities with named entity recognition techniques. Further, to improve the inference recall, we construct a Delivery Mobility Graph and spatial ranking model driven by couriers' reporting behaviors.
- Based on the evaluation over three large-scale real-world datasets, `CoMiner` achieves an average accuracy of 95.1% and outperforms state-of-the-art methods by 20.3%. Moreover, `CoMiner` has been deployed at JD Logistics for six months and has inferred coordinates for over 30 million addresses with an accuracy of 93.3%, i.e., within 300m of the ground truth coordinates [33].
- We have deployed `CoMiner` to reduce the number of parcel re-routings by an average of 910 every day, which saved more than one million RMB (~145,600 US $) for the company in one year. Further, we design, develop, and deploy an *Abnormal Delivery Detection System* in the Chinese City *Hefei*, which improves the detection of abnormal delivery events by 11.6%. The detected abnormal delivery events have been utilized for decision-making to improve user experience on the platform.

2 BACKGROUND AND MOTIVATION

In this section, We first introduce the preliminaries and motivations of the coordinate inference problem in the E-commerce logistics setting. Then, we conduct data-driven investigations on real-world datasets to show the problem's importance and challenges.

2.1 Preliminaries and Problem Formulation

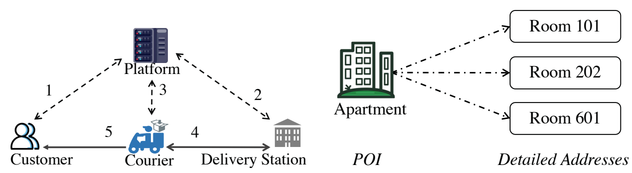
We first define address, POI, and waybill, which will be used to describe the parcel delivery process in E-commerce logistics.

Definition 1. *(**Address**) (Addr) is created by users to describe a physical location in a text format.*

Definition 2. *(**Point-of-Interest**) (POI) A POI represents a spatial entity that interacts with citizens by providing a specific urban function. A POI normally contains multiple Addr.*

Definition 3. *(**Waybill**) is the description of an order delivery task that created by the E-Commerce platform, which is denoted as*

DOrder = (Addr,courierID,orderID,status), where $status$ is an order status (e.g., in delivery), $courierID$ and $orderID$ are the unique ID for a courier and an user placed order, respectively.



(a) Order Delivery Process. (b) POI and Address Relationship.

Figure 1: Delivery Process (a) and POI-Address Example (b).

A Common Parcel Delivery Process. In Fig. 1(a), we illustrate the E-Commerce parcel delivery process, which integrates three main stakeholders, i.e., users, couriers, and the E-Commerce platform. The parcel delivery process consists of five steps. (1) The user places an online order with an address on the platform, and the platform creates a waybill (see Def. 3, containing text address, order ID, etc.). (2) The platform infers the order destination coordinates by Geocoding the textual address and transmits the order parcel to the corresponding delivery station. (3) The platform assigns the waybill to a courier. (4) The courier picks the parcel up at the delivery station with a mobile device, i.e., PDA. (5) The courier locates the POI (see Def. 2) and then drops the parcel off at the user's address (see Def. 1). The courier will carry the PDA during the delivery process and the PDA will generate delivery trajectories (see Def. 4). The courier will then manually report a delivery event (see Def. 5) when he/she successfully drops the parcel off at the user's address.

Definition 4. ***(Delivery Trajectory)*** *($Traj_{id}$) is a sequence of spatial temporal points generated by a GPS-enabled device carried by the courier while delivering parcels, denoted as $Traj_{id}$ = <p_1, p_2, ..., p_i, ... , p_n>, where p_i = (lat_i,lng_i,t_i), representing the latitude, longitude, timestamp, and id is the ID of a courier.*

Definition 5. ***(Delivery Event)*** *($Deli$) is an action performed by the courier when a parcel is delivered to user's address, denoted as $Deli = (lat, lng, t, orderID)$, representing the parcel $orderID$ is delivered at timestamp t at location (lat, lng).*

In the parcel delivery process mentioned above, a courier delivers parcels by first locating the POI of the corresponding parcel address. A POI can represent multiple addresses spatially close to each other inside the POI, i.e., share similar geographic coordinates. For example, in Fig. 1(b), an apartment is a POI, and each room in the apartment is an address of this POI. Knowing the geographic coordinates of the POI can satisfy the delivery task because addresses can be located by couriers with instructions and local maps at the POI. Therefore, we focus on inferring the coordinates of POIs to represent the coordinates of addresses, which have practical significance in logistics delivery.

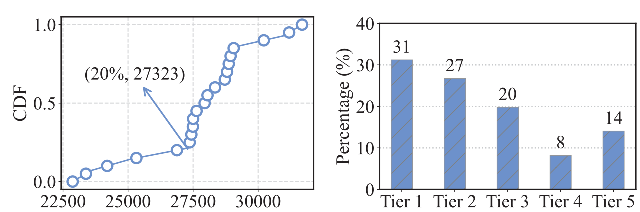
2.2 Data-driven Investigation

We first introduce the dataset used for data-driven investigations. The multi-modal dataset is collected by JD Logistics during couriers' parcel delivery. The data is entered by users or uploaded by couriers' PDA, which consists of three parts, i.e., delivery trajectory data, waybill data, and delivery event data. The delivery trajectory includes key features such as latitude and longitude at corresponding timestamps; Waybill data includes the textual address, parcel status, and other order information; Delivery event data includes the coordinates and timestamp when a courier reports parcel delivery. We show an example in Table 1.

Table 1: An Example of the Multi-modal Dataset.

Delivery Trajectory	lat	lng	time	hours	courierID
	37.50	121.39	11-12 08:00:30	8	210043
Waybill	status	station	address	time	orderID
	150	4203	Yantai City, Lai Shan, Nan Park	11-12 09:30:30	110156 320198
Delivery Event	lat	lng	deliveryNote	time	orderID
	37.50	121.39	Parcel delivery	11-12 09:30:30	110156 320198

Based on the dataset, we conduct data-driven investigations to show the importance of coordinate inference. Inaccurate coordinates of the E-commerce logistics platform will lead to parcel re-routing because of dispatching parcels to the wrong delivery stations. Such re-routing will increase couriers' delivery distance (i.e., couriers need extra effort to deliver a re-routing parcel) and the platform's cost. For example, in Fig. 2(a), the number of daily re-routing is more than 27,323 in 80% days, which costs the E-Commerce logistics platform 4.6 million US dollars each year ($0.45/parcel re-routing). In addition, in Fig. 2(b), we find that re-routing exists in many cities across the country, especially in large cities (31% of parcel re-routing is in Tier 1 cities of China, e.g., Beijing and Shanghai), which indicates the significance of accurate coordinate inference to reduce the economic and environmental costs.



(a) Daily Number of Re-routing. (b) Re-routing in Different Cities.

Figure 2: Parcel Re-routing Frequencies (a) and Fraction of Re-routing among Different Tiers of Chinese Cities (b).

2.3 Challenges

However, it is not trivial to infer coordinates for addresses based on the data in Table 1 due to the following two challenges:

(i) Data sparsity in delivery. Due to users' shopping pattern variance (i.e., frequency), different addresses have a different number of physical visits by couriers and thereby have a different number of delivery-related data. Fig. 3 illustrates the number of orders for each address in a delivery region from Jan. 2020 to Jan. 2021, where 71% of addresses have only 1 or 2 orders in one year, which leads to a data sparsity issue to infer the coordinates of these addresses.

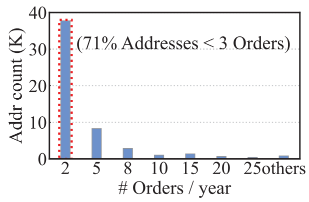


Figure 3: Number of Orders per Address in one Year.

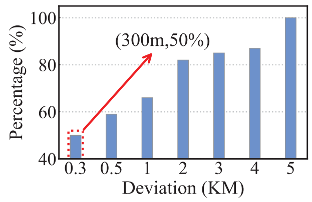


Figure 4: Deviation between $Coor_{Deli}$ and true $Coor_{addr}$.

(ii) Uncertain reporting behavior. Due to the reporting behavior variance of couriers, some couriers report delivery events at the wrong places or wrong timestamps. The main reasons for the existence of uncertain reporting behavior are two folds. Firstly, the courier forgets to report delivery when he/she delivered the parcel because of the heavy delivery load during peak hours. Secondly, the courier intentionally reports delivery before they deliver the parcel for orders with strict time deadlines. This challenge is non-trivial because we cannot correct couriers' reporting behavior even with the real-time uploaded trajectories and derived stay time for reported locations, i.e., we do not know whether the reported location is the actual location of the corresponding address. Fig. 4 shows the CDF of deviation between delivery coordinates and the true coordinates from a delivery region. Only 50.3% of delivery coordinates deviate from the ground truth by less than 300 meters. Therefore, taking the GPS coordinates of delivery events as the addresses' true coordinates will bring a large deviation error.

In summary, motivated by the importance and challenges of the coordinate inference problem, we aim to design a framework that can automatically infer coordinates for large-scale textual addresses considering both data sparsity and uncertainty.

3 COMINER DESIGN

Fig. 5 describes the coordinate inference process based on three key components, i.e., (i) A POI-Level Clustering module (Sec. 3.1) takes couriers' reported coordinates with corresponding addresses as input, and infers addresses' true coordinates based on coordinate clustering. This clustering model can only infer coordinates for partial addresses due to the data sparsity issue. Thus, we then design a (ii) Mobility Graph Construction module (Sec. 3.2) that uses addresses as nodes and couriers' address visiting as edges to organize both ***known addresses*** (coordinates have been inferred) and ***unknown addresses*** (coordinates to be inferred) in the same graph. Based on the constructed graph, we design a (iii) Behavior-Driven Address Ranking module (Sec. 3.3) to infer coordinates for *unknown addresses* considering couriers' reporting behaviors.

3.1 POI-Level Clustering Model

In this subsection, we introduce the POI-level clustering model. Specifically, we compare address-level clustering with POI-level clustering and design a density-based clustering model based on POIs identified from textual addresses.

Address-Level Clustering. Intuitively, each address has multiple delivery coordinates $Coor_{Deli_1}, ..., Coor_{Deli_n}$ generated by couriers after delivering parcels at the address. A straightforward method is to cluster these delivery coordinates and the center of the cluster

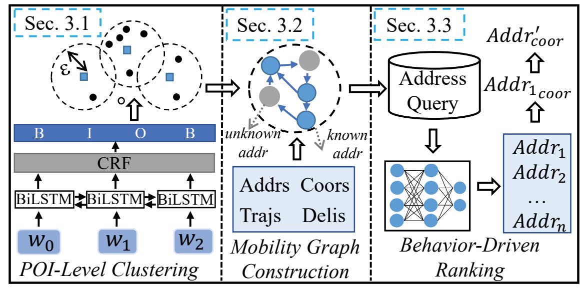


Figure 5: Framework of CoMiner.

is the inferred coordinate for an address. However, as in Fig. 3, most addresses have a limited number of delivery coordinates. This data sparsity challenge will decrease the performance of coordinate inference, i.e., the clustering-based model will not generate a cluster representing the address coordinates with high confidence.

POI-Level Clustering. Compared with the address, a POI has more orders and visits because each POI has multiple associated addresses (see Sec. 2.1). We further compare users' shopping pattern of POI level with address level and have an important finding as in Fig. 6: 80% of POIs have more than three times of orders than addresses, i.e., more delivery visits by couriers. Therefore, the POI-level clustering model will bring better performance and solve the data sparsity challenge. Based on this data-driven finding, we design a POI-level clustering method for coordinate inference and the inferred coordinates will be assigned to addresses in the POI. However, it is not feasible to extract POIs from addresses using rule-based models due to the complexity of the Chinese addresses.

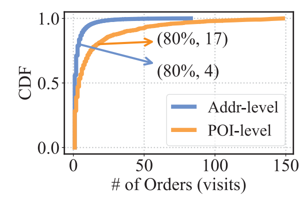


Figure 6: Number of Orders.

To solve this problem, we formulate POI identification as a sequence labeling task, and design POI-NER, a Named Entity Recognition (NER) model to tag POIs in addresses. The POI-NER model consists of three LSTM layers and a Conditional Random Field (CRF) layer based on a popular model [17, 19]. Even though there are lots of public NER datasets and models, we cannot directly apply them to our problem because the component of textual address is significantly different from general text data such as Wikipedia. Thus, we investigate the components of addresses and design six types of labels, i.e., *PROVINCE, CITY, DISTRICT, ROAD, POI, O (oral language)*. The POI-NER model is trained by maximizing the conditional log-likelihood as in Eqn. (1).

$$\theta_p = \arg\max_{\theta_p'} \sum_{i=1}^{N} (y_i|w_i, \theta_p') \tag{1}$$

where θ_p' represents all trainable parameters in POI-NER, N is the number of training addresses, w_i is a sequence of words in the i_{th} address, y_i is the corresponding label sequence for w_i. Given a new address, the POI-NER labels and extracts POIs based on Eqn. (2), where $\hat{y}$ is the predicted label sequence for w. For example, $\hat{y}$ = *["B-PROVINCE", "I-PROVINCE", "B-CITY", ..., "B-POI", "I-POI", ...]*, where *B-POI* represents the word is the beginning of a *POI* entity", *I-POI* represents the word is inside a *POI* entity.

$$\hat{y} = \arg\max_{y} p(y|w, \theta_p) \tag{2}$$

After the POIs are identified by the POI-NER model, we construct a POI-coordinate dictionary for POI-level clustering. The key is POI and the value is the set of delivery coordinates generated by couriers when dropping parcels off at the POI. Intuitively, couriers normally deliver parcels at a fixed location of each POI, and the location with dense delivery coordinates has a higher probability to be the true location of a POI. Thus, we apply the density-based model DBSCAN [11] to cluster the coordinates. DBSCAN generates multiple clusters for each POI by clustering its delivery coordinates. The average coordinates of all points in the densest cluster (i.e., with the maximum number of points) are assigned as the POI's coordinate, and assigned as the coordinate for all addresses inside of the POI. The DBSCAN distance threshold ϵ is 0.0003 degrees and the minimum number of points in a cluster is 5.

3.2 Mobility Graph Construction

Even though the POI-level clustering model can infer coordinates for POIs and addresses with high precision, it will fail for two scenarios: (i) Only a few orders are delivered at a POI in a period of time because reasons such as the user uses aliases (different names) of POI or the POI is of small scale, e.g., a convenience store; (ii) Many orders are delivered at a POI but most of the delivery coordinates significantly deviate from each other and thereby a dense cluster cannot be generated by the clustering model, i.e., coordinates cannot be inferred. Thus, we need another module to infer coordinates for addresses that cannot be inferred by the POI-clustering model (remaining coordinates).

Intuition for Remaining Coordinate Inference. To fill this gap, as in Fig. 7, we consider a delivery sequence $<Deli_A, Deli_B, Deli_C>$ at addresses $<A, B, C>$ with increasing delivery time. We categorize the pattern of delivery sequences into three classes. (1) $Deli_A$, $Deli_C$ are normal delivery events. (2) Only one of $Deli_A$ and $Deli_C$ is a normal delivery event. (3) Both $Deli_A$ and $Deli_C$ are abnormal delivery events. The abnormal and normal delivery events are defined in Def. 6. Intuitively, the courier delivers parcels one by one and reports delivery events following the parcel delivery sequence. Therefore, we can infer the coordinates of B if the delivery events at A and C are normal (class (1)) because B lies on the delivery trajectory between A and C. As a result, by utilizing this certain pattern of couriers' delivery behavior, we can infer the coordinate for B, which cannot be inferred by the POI-level clustering model. Note that, even though the delivery events at A and C might not be normal delivery events (class (2) or (3)), we can still identify a class (1) delivery sequence on other days because address B normally has a few orders in a year.

Definition 6. ***(Abnormal/Normal delivery event)*** *A delivery event is abnormal if* $dist(Coor_{Deli}, Coor_{Addr}) > \Delta d$*, i.e., the distance between the reported delivery coordinates* $Coor_{Deli}$ *and the true coordinates* $Coor_{Addr}$ *(groundtruth) is greater than a threshold* Δd*. A delivery event is normal if* $dist(Coor_{Deli}, Coor_{Addr}) < \Delta d$*.*

Based on the above intuition, we design a delivery mobility graph. The courier visits a group of addresses for parcel delivery

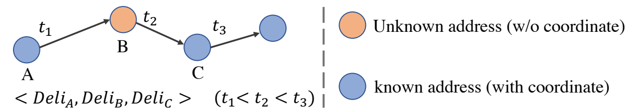


Figure 7: Delivery Sequence Illustration.

each day starting from the delivery station. The visiting addresses and the visiting sequences among addresses vary on different days. Therefore, the courier's mobility trace between addresses can be considered as a time-varying graph (TVG) [8], which is defined as *Delivery Mobility Graph*.

Definition 7. ***(Delivery Mobility Graph (DMG))*** *A DMG is represented by* $DMG = (V, E)$*, where* V *is a set of nodes* $v_1, v_2, ..., v_n$*, each node represents an address with corresponding delivery time, and* $E = \{(v, v')|(v, v') \in V \times V\}$ *is a set of directed edges from* v *to* v'*, each edge represents a courier's move from node* v *to node* v'*.*

We describe the *DMG* construction process in detail. Based on the POI-level clustering model, some addresses' coordinates have been inferred (*known addresses*). Thus, for each node in *DMG*, we have three features, i.e., a flag representing an address is known or unknown, true coordinates, and reported coordinates by couriers. Given delivery events, trajectories, addresses, and time interval threshold ΔT, we add the edge from the delivery station to the first visited node $V[1]$. Then, we iterate through all visited nodes and add an edge if the duration between two visits is less than ΔT. Nodes with a duration longer than ΔT will terminate the graph construction process because nodes are visited by couriers in two delivery trips (delivery trip is the courier's delivery process without returning back to the delivery station), e.g., in the morning and afternoon, respectively. We construct *DMG* for each delivery trip. Note that, the *DMG* is a linked list if all addresses in a delivery trip are only visited once, which is a special case of the graph.

Now we have both *known addresses* (with true coordinates) and *unknown addresses* (with couriers' reporting coordinates), the delivery trajectories connecting *known* and *unknown addresses*, the true coordinates of *unknown addresses* is on the trajectories. In the next step, we design a ranking model to infer coordinates of remaining *unknown addresses* based on the constructed Delivery Mobility Graph (*DMG*) and couriers' reporting behavior.

3.3 Behavior-Driven Ranking Model

As in Fig. 8, our ranking model has four steps: (i) candidate stay point generation, (ii) semantic representation of stay points, (iii) stay point ranking, and (iv) address coordinates generation.

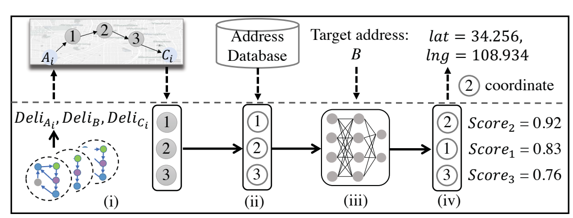


Figure 8: Behavior-Driven Ranking Model.

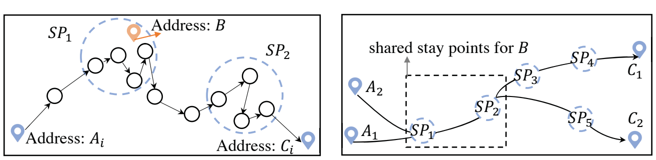
Intuition. Even though couriers' reporting behavior is uncertain, we identify a regular pattern under the uncertain behaviors. A courier reports delivery events according to the consecutive delivery sequences, i.e., the order of reporting is consistent with the order of actual delivery. Therefore, we can probably address the reporting uncertainty challenge by utilizing the coordinates of *known addresses* and delivery sequences on the Delivery Mobility Graph.

Candidate Stay Point Generation. The step (i) in Fig. 8 illustrates the process of candidate delivery sequence generation from the *DMG*, and the candidate stay points (①②③ in Fig. 8) generation process based on delivery sequences. On the *DMG*, we represent all nodes with unknown coordinates as B and nodes with known coordinates as A (if $A_t < B_t$) or C (if $B_t < C_t$). The objective is to infer the coordinates of B (B_{coor}) by utilizing A and C. For each unknown node B, we first perform a graph search to obtain candidate delivery sequences ($Deli_{seq}$) $\{..., < Deli_{A_i}, Deli_B, Deli_{C_i} >, ...\}$ under the time interval constraint Δt, i.e., the duration of <$Deli_{A_i}$,$Seli_B$> and <$Deli_B$,$Seli_{C_i}$> is less than Δt.

We then detect candidate stay points (i.e., *a group of consecutive points in a trajectory that generated during a stay of a moving object at a location*) on the delivery sequences between A_i and C_i as shown in step (i) of Fig. 8. Intuitively, the courier will stay for some time when he/she visits an address for parcel delivery, which will potentially generate stay points. The coordinates of textual addresses have a high probability to be close to one of these stay points. Thus, we can detect couriers' stay points from trajectories and infer addresses' coordinates by selecting the best stay point, which is more efficient and effective than selecting the best GPS point in trajectories.

Fig. 9 introduces the candidate stay point generation process on delivery sequences. Specifically, stay points between A_i and C_i of all sequences in $Deli_{seq}$ are detected by applying the stay point detection algorithm [22], such as SP_1 and SP_2 in Fig. 9(a). Then, Fig. 9(b) illustrates the candidate stay points generation process by calculating the spatially shared stay points S_{points} for all <A_i, B, C_i> sequences, i.e., B appears in many delivery sequences on the *DMG* and thereby the shared stay points have higher probabilities to contain address B. For example, SP_1 and SP_2 are detected as candidate stay points for address B because they are both in <A_1, B, C_1> and <A_2, B, C_2>. All stay points will be candidates if no shared stay points in <A_i, B, C_i>. Note that not all candidate stay points are caused by the delivery and can represent addresses' coordinates because couriers' other activities such as waiting for traffic lights or jams also generate stay points. Thus, the problem is re-formulated as how to design a model $\mathcal{F}$ to select the best stay point S_{best} among S_{points} to represent the B's coordinates B_{coor}:

$$B_{coor} = \mathcal{F}([\ SP_i | SP_i \in S_{points}]) \tag{3}$$



(a) Stay point detection in trajectories (b) Candidate stay point generation

Figure 9: Illustration of Candidate Stay Point Generation.

Stay Point Ranking. We aim to represent the coordinates of address B by the stay point that contains B through a ranking model.

Table 2: Address Similarity Example.

B	*Xi'an City, Yanta District, Jinshui Tower Health Club*
B' in SP_k	*Xi'an City, Yanta District, Jinshui Building*

However, there is no information to measure the similarity between stay points and unknown addresses, which makes the ranking model not trivial. The POI-level clustering model has inferred coordinates for some addresses and constructed the *<address, coordinates>* database, which can potentially provide similarity information for stay points. Thus, as the step (ii) in Fig. 8, we query the database with center coordinates of stay points and get addresses near stay points to get S'_{points}, which contains nearby textual address information and can be used to calculate similarities. For example, Table 2 introduces an unknown address B (a club) and a similar address B' (a building containing the club) in a candidate stay point SP_k. We aim to rank SP_k as the top one stay point and assign SP_k's coordinates to B.

To rank all stay points in S'_{points} with low computational cost, we design a ranking model based on the longest common subsequence detection. As shown in the step (iii) of Fig. 8, we compute the longest contiguous subsequence between target address B with each address in all stay points to represent the similarity, i.e., a longer subsequence means a higher similarity score between B and the corresponding stay point. Then, the stay points in S'_{points} are ranked according to their similarity scores, and the coordinates of the stay point with the highest similarity score will be assigned to B as shown in step (iv) of Fig. 8.

4 EVALUATION

In this section, we introduce the evaluation setup followed by an extensive evaluation of `CoMiner` compared with baseline models. We also introduce real-world deployment and two applications.

4.1 Evaluation Setup

Evaluation Dataset. To evaluate `CoMiner`, we utilize three real-world delivery datasets from JD Logistics, which cover three delivery regions in Hefei, Beijing, and Anqing city, respectively. These datasets have 313,026 addresses with labeled coordinates, one-year delivery events, trajectories, and waybills of 88 couriers from Jan. 2020 to Jan. 2021. The delivery-related data is collected when the courier is delivering parcels. The ground truth coordinates are labeled by domain experts utilizing multi-source data, e.g., POI boundary data, map data, and platform transaction data.

Evaluation Baselines.

- DeliEvent assigns the delivery coordinate to the address by assuming the delivery event is accurate.
- TextMatch [25] calculates the similarity between a given address with all addresses in the same region. The coordinate of the candidate with the highest similarity score is assigned as the coordinate for the given address.
- GeoCloud [33] utilizes delivery data to infer coordinates for customers' addresses. A cluster-based method is proposed for coordinate inference.

Table 3: Baseline Comparison for Accuracy and Recall.

Model	Hefei		Beijing		Anqing	
	Acc(%)	R(%)	Acc(%)	R(%)	Acc(%)	R(%)
DeliEvent	65.7	100	58.3	100	70.8	100
TextMatch[25]	62.5	88.2	75.9	82.7	71.3	78.8
GeoCloud[33]	92.2	39.0	96.8	69.4	92.6	45.4
DTInf[31]	77.0	**100**	74.0	**100**	73.3	**100**
CoMinerAVG	57.7	98.4	74.0	95.8	65.3	97.1
`CoMiner-`	90.1	**100**	96.6	**100**	91.9	**100**
`CoMiner`	**93.1**	85.9	**97.3**	91.4	**95.0**	84.5

* Acc refers to 300m Accuracy, R refers to Recall.

- DTInf [31] is originally designed for delivery time inference from the delivery trajectory in logistics. It is also able to infer the GPS coordinates for addresses based on delivery trajectories.
- CoMinerAVG calculates the average coordinate of A and C in the delivery sequence $< A, B, C >$ and assigns it to B.
- `CoMiner-` is a variant of `CoMiner` (i.e., `CoMiner-`), where all coordinates can be inferred. For the coordinates that cannot be inferred by `CoMiner`, we apply address matching via computing textual similarity with already inferred addresses.

Evaluation Metrics. We evaluate all models by accuracy (*Acc*) and recall (*Recall*). *Acc* of i-meter is the fraction of inferred coordinates with a deviation error less than i-meter to the true coordinate. $Recall = \frac{n}{N}$, where n is the number of addresses with coordinates inferred successfully, and N is the number of total addresses. There is a trade-off between *Acc* and *Recall* for similarity or cluster-based models, e.g., the improvement of *Acc* by increasing the similarity threshold would decrease the *Recall*.

Evaluation Granularity. We investigate the *Acc* and *Recall* on five granularities, i.e., 100m (100-meter), 200m, 300m, 500m, and 1000m. We mainly focus on 300m to compare different models following the common industrial standard [33]. The experimental result is for 300m by default if there is no specific description.

4.2 Data-driven Offline Evaluation

In this subsection, we describe the offline evaluation results and investigate the impact of multiple factors on the model performance.

Overall Performance. From the evaluation results in Table 3, we conclude several important findings: (i) `CoMiner` achieves the highest *Acc* and competitive *Recall* over all cities. Even though *GeoCloud* [33] achieves a similar high *Acc*, it has a low *Recall* of 47.3% compared with our result (84.5%). The main reason is that some coordinates cannot be inferred because of the uncertain number of visits, i.e., some addresses' have few orders and thus the coordinates cannot be clustered and inferred. (ii) `CoMiner`'s variance `CoMiner-` achieves 100% of *Recall* and high *Acc*. `CoMiner-` has significantly higher *Acc* than the other two models *DeliEvent* and $DTInf$, which also have 100% of *Recall*. In detail, it is not surprising that *DeliEvent* has low *Acc* due to couriers' uncertain reporting behavior. The main reason for the low *Acc* of *TextMatch* is the complex characteristics of Chinese addresses, i.e., two addresses have high textual similarity but are geographically far from each other. $DTInf$ first infers coordinates based on delivery caused stay points and then applies a hierarchical cluster model to refine the inferred coordinates. The loss of *Acc* is mainly caused by the highly uncertain characteristics of delivery data, e.g., couriers only have a short stay time at some addresses.

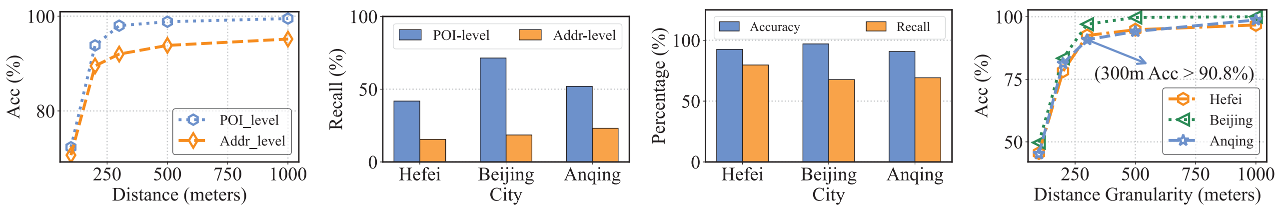
There are two main reasons for the loss of *Recall* of `CoMiner`, i.e., some addresses' coordinates are not successfully inferred (14.1% in Hefei, 8.6% in Beijing, and 15.5% in Anqing City, respectively.) (i) Delivery trajectories collected from the real world are not consistently of high quality. As a result, the stay point cannot be detected in sparse trajectory segments. (ii) The inference will also fail when there is no consecutive delivery sequence $<A, B, C>$ extracted from *Delivery Mobility Graph* given the time interval Δt. For example, when B is the first or last delivery location in the delivery process. However, we can easily extend `CoMiner` to `CoMiner-` for scenarios where *Recall* is the priority.

Performance of POI-Level Clustering Model. Fig. 10(a) and 10(b) show the *Acc* and *Recall* of POI-level and Address-level clustering models, respectively. With the increase of distance granularity, the POI-level model slightly outperforms the address-level model even though both of them achieve high *Acc*. The reason is that at the POI level, more delivery coordinates are considered in the clustering process, which leads to higher *Acc*. Fig. 10(b) shows that the *Recall* of the POI-level cluster model is significantly higher than the Address-level model (26.3%~52.8%), which is because we can identify more addresses at the POI level due to more delivery visits to a POI than to an address in the POI.

Impact of City on Ranking Model. Fig. 10(c) illustrates the model robustness by comparing the results in three cities. `CoMiner` shows similar performance even though the areas and geographic characteristics of the three cities significantly differ from each other (e.g., different traffic conditions). The 300m *Acc* across three cities ranges from 90.8% to 97%. The *Recall* varies from 67.7% to 79.7%. *Hefei* has the highest *Recall*, which is consistent with Fig. 11(d). The main reason might be that *Hefei* city has a higher ratio of accurate delivery events, which increases the number of delivery sequences and brings higher $recall$. We can also observe that even though there are potentially more GPS drifts causing GPS inaccuracy in big cities such as Beijing, `CoMiner` still has consistent performance over three cities, which demonstrates the robustness of our clustering and behavior-driven ranking algorithm.

Impact of Distance Granularity on Ranking Model. Even though 300m *Acc* is the most important metric, we are also interested in how `CoMiner` performs with different distance granularities, i.e., 100m~1000m. As shown in Fig. 10(d), the *Acc* increases with the distance granularity. `CoMiner` yields a 300m *Acc* higher than 90.8% on all stations. 100m *Acc* ranges from 45.2% to 49.7% (not plotted in Fig. 11 due to space constraint). The main reason for the low 100m *Acc* lies in two folds: (i) A POI (e.g., school) can cover a large area and the distance between the POI center to its boundary is more than 100-meter, which would significantly decrease the average 100m *Acc*. (ii) GPS coordinates could shift away from the true locations in areas with tall buildings.

Impact of Time Interval on Ranking Model. We investigate how time interval Δt, a hyper-parameter to extract delivery sequences, impacts *Acc* and *Recall*. Intuitively, a smaller Δt brings fewer coordinate candidates and thereby leading to higher *Acc* and lower



(a) POI-Address Level Accuracy. (b) POI-Address Level Recall. (c) Accuracy and Recall. (d) Different Distances.

Figure 10: (a) (b) POI-Address Level Comparison. (c) (d) Performance of `CoMiner` on 3 Stations.

Recall. We present the impact of Δt to *Acc* in Figs. 11(a), 11(b), 11(c), and to *Recall* in Fig. 11(d). We note that with the increase of Δt, 200m *Acc* drops significantly while 300m, 500m, and 1000m *Acc* are relatively stable. The main reason is that the inferred coordinates already have high *Acc* at 300m, 500m, and 1000m, and thereby will not have a significant increase with Δt. In Fig. 11(d), *Recall* of all datasets increases with Δt, which is because larger Δt leads to more inferred coordinates, i.e., higher *Recall*.

4.3 Online Deployment and Applications

In this subsection, we introduce the online deployment and two downstream applications of `CoMiner`. One technical metric and two business metrics are designed to measure the framework performance. Since geographic coordinates are one of the key data elements in location-based services, the company we work with has a strong incentive to infer accurate coordinates for addresses and POIs to enable various downstream applications, e.g., re-routing reduction, and abnormal delivery event detection.

Deployment Setup. We adapt a two-phase deployment mechanism, i.e., first, deploy `CoMiner` at Beijing City, and then deploy `CoMiner` to the whole nation if the online *Acc* satisfies the threshold (>90%). To enable streaming coordinate inference, we decompose our framework into two modules: (i) the data preprocessing module, which is responsible for data acquisition from the company's big data platform and data preprocessing; (ii) the coordinate inference module, which is for coordinate inference and saving the mined results into the cloud database. These two modules are deployed in the company's cloud server to work simultaneously in a streaming manner. The server is equipped with Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, 8 cores, 80GB RAM with Python 3.6.

Evaluation 1 (technical metric): Deployed Model Accuracy. To evaluate the online performance of our model, we randomly sample addresses with inferred coordinates from the database to manually evaluate the 300m *Acc* in two deployment phases. In phase 1, we randomly sample and evaluate 1000 addresses after one week's deployment and achieve 94.5% of *Acc*. In phase 2, we randomly sample and evaluate 1000 addresses twice and achieve *Acc* of 92.3% and 93.1%, respectively.

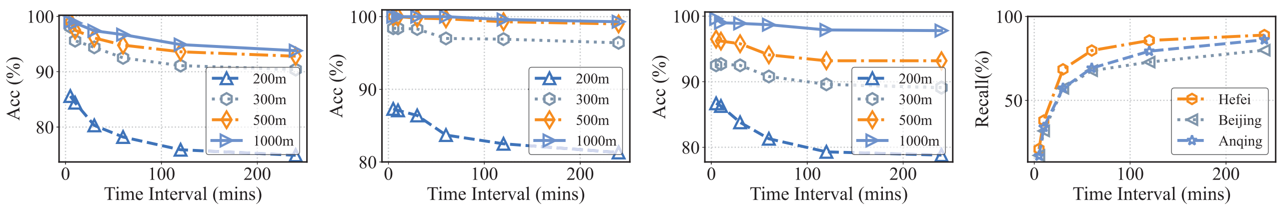
Evaluation 2 (business metric): Re-Routing Reduction. The number of abnormal parcel re-routing is an important business metric to evaluate the company's order dispatching system because re-routing has caused great loss to the company (money cost), couriers (extra delivery distance), and users (extra parcel waiting time). Based on `CoMiner`, we update the company's order dispatching system with addresses and inferred coordinates. We conduct an online A/B test to evaluate the performance of the new order dispatching system in terms of the daily re-routing number. As the result in Fig. 12(a), we witness an average of 910 re-routing reductions every day (i.e., 3.3% of all the parcel re-routings) from August 1st to 31st, 2021, which will lead to an annual loss reduction of about one million RMB (i.e., 2.9 RMB per re-routing on average).

Evaluation 3 (business metric): Abnormal Delivery Event Detection. The abnormal delivery event in last-mile delivery hurts the user experience because the user would feel cheated when he/she receives a notification *"Your parcel is delivered"* while waiting for the parcel. Therefore, another important business metric is the detection rate of abnormal delivery events. We develop and deploy a `CoMiner`-based abnormal delivery detection system in a Chinese city, Hefei, with 12 delivery regions and more than 200 couriers. For privacy protection, our system is designed to evaluate the abnormal delivery rate at the aggregate level (i.e., delivery region) rather than at the courier level. Fig. 12(b) illustrates the detection result from July 19th, 2021 to July 25th, 2021 with an average detection rate of 11.6%. Our system can detect 60,500 and 22 million abnormal delivery events in one day and one year in Hefei city, respectively. The detection result has been utilized for the company's decision-making to improve user experience, e.g., giving extra bonuses for delivery regions with lower abnormal delivery rates.

5 DISCUSSION

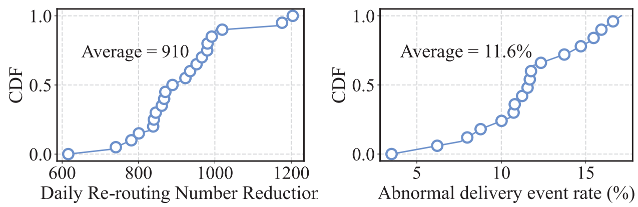
5.1 Lessons Learned

- **Data-Driven Findings**. We have two new findings based on the study of delivery data. (i) The commonly used commercial maps do not perform well for complex and large-scale Chinese shipping addresses. (ii) Some couriers tend to report delivery earlier than real delivery while some tend to report later, even though they are supposed to report delivery accurately in the design of the delivery system (Fig. 4).
- **Impact of Human Behavior to Online-to-Offline (O2O) System Design.** In the real world, due to uncertain physical elements such as weather and delivery task workload, human behavior might differ from the system designer's expectation. Considering such human behavior in system design will benefit the system's performance. By modeling couriers' uncertain reporting behavior, the coordinate inference performance is significantly improved compared with baseline models (Table 3).



(a) Accuracy of the Hefei City. (b) Accuracy of the Beijing City. (c) Accuracy of the Anqing City. (d) Recall of the three Cities.

Figure 11: Impact of Time Interval on Accuracy and Recall of the Ranking Model.



(a) Reduction of Re-Routing. (b) Detection Rate.

Figure 12: Re-Routing Reduction (a) and Abnormal Delivery Event Detection (b).

5.2 Privacy Protection

We take four concrete steps to protect user privacy in the data used in this study. (i) The data is collected only during work hours under the consent of couriers for purposes of parcel insurance and academic research, etc. (ii) The couriers can turn off the GPS module at any time to stop uploading trajectory data. (iii) All user identifiers are removed or replaced with a random number. (iv) The data is only accessed by the core team members who have signed non-disclosure agreements of the data.

6 RELATED WORK

Our study focuses on automatically inferring GPS coordinates for given addresses. We categorize the related work into two categories, i.e., Geocoding (GC) and trajectory data mining.

Geocoding (GC). Geocoding, which associates texts with geographic coordinates, has been widely studied in recent years. One group of GC work requires no extra effort but cannot update new information timely [6, 20, 28, 29, 37, 39]. These studies utilize open-source datasets and machine learning models to build local GC services or to improve GC algorithms. Even with the advantage of economic and low privacy concerns, they suffer from low recall and not being able to update POI or address information. Kulkarni et al. [20] propose a multi-level geocoding model(MLG) to map text into geographic coordinates on three public English datasets. Chatterjee et al. [9] utilize map data as the reference data source for GC. One kind of GC service that can timely update new address data is the commercial GC service such as Baidu Map [2]. These services require extensive extra labor and devices (e.g., street view cars) for collecting and updating new POIs and addresses. There are also open-source GC systems [4, 5] that are free to use, but with the limitation of being inaccurate [43].

Thus, the ideal GC service is economic, accurate, and fast. Srivastava et al. [33] utilize E-Commerce delivery data in India to build a GC service. This paper assumes lots of delivery events for each location and applies DBSCAN [11] to obtain GPS coordinates. However, their method is not suitable for our scenario since some addresses have very few delivery events. In on-demand delivery, Song et al. [32] propose an image-based method for POI location correction where the data is collected from workers' phones and has good quality, which is different from the logistics scenario. Ruan et al. [31] propose DTInf to infer delivery time and can be applied to infer delivery coordinates automatically from delivery trajectories. Even with a high recall, DTInf suffers from low accuracy when the trajectory is of low quality or when couriers' reporting behavior is highly uncertain. These studies do not consider human behavior uncertainty in data collection and algorithm design, thereby can hardly achieve ideal performance at scale.

Trajectory Data Mining. Trajectory data mining focuses on preprocessing, managing, mining valuable knowledge, and design novel applications from trajectory data [7, 12–16, 24, 38, 40, 41, 44, 45]. Trajectory data preprocessing and managing have received much research interest in recent years. Tong et al. [36] propose a model to reconstruct vehicle trajectory based on mobility correlation and vision analysis. Ruan et al. [30] propose DeppMG to generate urban maps based on large-scale noisy trajectories. Li et al. [23] design a holistic distributed NoSQL trajectory data management framework to handle spatio-temporal data. Trajectory data benefits many applications. Nair et al. [27] categorize cycling trip GPS trajectories to understand urban cyclist behaviors. Based on vehicle trajectories, Li et al. [21] propose a deep learning model to predict urban traffic flow. Trajectory also has important applications in business. Zhang et al. [42] study the problem of billboard placement based on trajectory analysis. Unlike outdoor trajectories, Das et al. [10] collect indoor occupant moving trajectories to predict future trajectory in a room. Unlike existing trajectory mining studies, our work investigates uncertain human behavior in E-commerce parcel delivery and combines natural language processing with trajectory mining to design a coordinate inference model for large industrial applications.

7 CONCLUSION

In this work, we focus on the problem of automatically inferring GPS coordinates for textual addresses. Motivated by users' shopping patterns and couriers' reporting behaviors, we design `CoMiner`, a cost-efficient behavior-driven unsupervised coordinate mining framework. We implement `CoMiner` on a large-scale dataset from three Chinese cities, and the results show that `CoMiner` achieves

95.1% of inference accuracy. We then deploy `CoMiner` in JD Logistics, and it has inferred coordinates for more than 30 million addresses in six months. Further, `CoMiner` benefits two business applications, i.e., parcel re-routing reduction and abnormal delivery event detection.

8 ACKNOWLEDGMENTS

This work is partially supported by NSF 1849238, 1932223, 1951890, 1952096, 2003874, and 2047822. We thank all the reviewers for their insightful feedback to improve this paper.

REFERENCES

[1] 2020. Cross-border E-commerce LogisticsTrends. 2020. https://www.cnlogistics.com.hk/doc/CNL_intel_202011.pdf.
[2] 2021. Baidu Maps. 2021. https://map.baidu.com/.
[3] 2021. COVID-19 has reshaped last-mile logistics in 2020. https://www.weforum.org/press/2021/04/covid-19-has-reshaped-last-mile-logistics-with\-e-commerce-deliveries-rising-25-in-2020/.
[4] 2021. GeoNames. 2021. http://www.geonames.org/.
[5] 2021. OpenStreetMap. 2021. http://nominatim.openstreetmap.org/.
[6] Pavel Berkhin, Michael R. Evans, Florin Teodorescu, Wei Wu, and Dragomir Yankov. 2015. A New Approach to Geocoding: BingGC. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, Washington) *(SIGSPATIAL '15)*. Association for Computing Machinery, New York, NY, USA, Article 7, 10 pages. https://doi.org/10.1145/2820783.2820827
[7] Edward Buckland, Egemen Tanin, Nicholas Geard, Cameron Zachreson, Hairuo Xie, and Hanan Samet. 2021. Managing Trajectories and Interactions During a Pandemic: A Trajectory Similarity-based Approach (Demo Paper). In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems.* 423–426.
[8] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. 2012. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems* 27, 5 (2012), 387–408.
[9] Abhranil Chatterjee, Janit Anjaria, Sourav Roy, Arnab Ganguli, and Krishanu Seal. 2016. SAGEL: Smart Address Geocoding Engine for Supply-Chain Logistics. In *SIGSPATIAL'16.* ACM, Article 42, 10 pages.
[10] Anooshmita Das, Emil Stubbe Kolvig-Raun, and Mikkel Baun Kjærgaard. 2020. Accurate Trajectory Prediction in a Smart Building Using Recurrent Neural Networks. In *UbiComp-ISWC'20.* ACM, 619–628.
[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD'1996 (KDD'96).* AAAI Press, 226–231.
[12] Joachim Gudmundsson, Martin P Seybold, and John Pfeifer. 2021. On Practical Nearest Sub-Trajectory Queries under the Fréchet Distance. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems.* 596–605.
[13] Baoshen Guo, Shuai Wang, Yi Ding, Guang Wang, Suining He, Desheng Zhang, and Tian He. 2021. Concurrent Order Dispatch for Instant Delivery with Time-Constrained Actor-Critic Reinforcement Learning. In *2021 IEEE Real-Time Systems Symposium (RTSS).* IEEE, 176–187.
[14] Baoshen Guo, Weijian Zuo, Shuai Wang, Wenjun Lyu, Zhiqing Hong, Yi Ding, Tian He, and Desheng Zhang. 2022. WePos: Weak-Supervised Indoor Positioning with Unlabeled WiFi for On-Demand Delivery. 6, 2, Article 54 (jul 2022), 25 pages. https://doi.org/10.1145/3534574
[15] Suining He and Kang G Shin. 2020. Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems. In *WWW 2020.* 88–98.
[16] Tianfu He, Guochun Chen, Chuishi Meng, Huajun He, Zheyi Pan, Yexin Li, Sijie Ruan, Huimin Ren, Ye Yuan, Ruiyuan Li, et al. 2021. POI Alias Discovery in Delivery Addresses using User Locations. In *SIGSPATIAL'21.* 225–228.
[17] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[18] Huiqi Hu, Yudian Zheng, Zhifeng Bao, Guoliang Li, Jianhua Feng, and Reynold Cheng. 2016. Crowdsourced POI labelling: Location-aware result inference and Task Assignment. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE).* 61–72.
[19] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
[20] Sayali Kulkarni, Shailee Jain, Mohammad Javad Hosseini, Jason Baldridge, Eugene Ie, and Li Zhang. 2021. Multi-Level Gazetteer-Free Geocoding. In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics.* Association for Computational Linguistics, Online, 79–88. https://doi.org/10.18653/v1/2021.splurobonlp-1.9
[21] Mingqian Li, Panrong Tong, Mo Li, Zhongming Jin, Jianqiang Huang, and Xian-Sheng Hua. 2021. Traffic Flow Prediction with Vehicle Trajectories. *AAAI* 35, 1 (May 2021), 294–302.
[22] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei ying Ma. 2008. Mining user similarity based on location history. In *In GIS '08: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems.* ACM, 1–10.
[23] Ruiyuan Li, Huajun He, Rubin Wang, Sijie Ruan, Tianfu He, Jie Bao, Junbo Zhang, Liang Hong, and Yu Zheng. 2021. TrajMesa: A Distributed NoSQL-Based Trajectory Data Management System. *TKDE* (2021), 1–1.
[24] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *WWW 2021.* 2177–2185.
[25] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, Vol. 6. 775–780.
[26] Mashaal Musleh, Sofiane Abbar, Rade Stanojevic, and Mohamed Mokbel. 2021. QARTA: an ML-based system for accurate map services. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2273–2282.
[27] Suraj Nair, Kiran Javkar, Jiahui Wu, and Vanessa Frias-Martinez. 2019. Understanding Cycling Trip Purpose and Route Choice Using GPS Traces and Open Data. *IMWUT.* 3, 1, Article 20 (March 2019), 26 pages.
[28] Benjamin J. Radford. 2021. Regressing Location on Text for Probabilistic Geocoding. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021).* Association for Computational Linguistics, Online, 53–57. https://doi.org/10.18653/v1/2021.case-1.8
[29] Sina Rashidian, Xinyu Dong, Shubham Kumar Jain, and Fusheng Wang. 2018. EaserGeocoder: Integrative Geocoding with Machine Learning (Demo Paper). In *SIGSPATIAL'18.* ACM, 572–575.
[30] Sijie Ruan, Cheng Long, Jie Bao, Chunyang Li, Zisheng Yu, Ruiyuan Li, Yuxuan Liang, Tianfu He, and Yu Zheng. 2020. Learning to generate maps from trajectories. In *AAAI*, Vol. 34. 890–897.
[31] 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 *KDD'2020.* 2813–2821.
[32] Yatong Song, Jiawei Li, Liying Chen, Shuiping Chen, Renqing He, and Zhizhao Sun. 2021. A Semantic Segmentation Based POI Coordinates Generating Framework for On-Demand Food Delivery Service. In *SIGSPATIAL'21.* ACM, 379–388.
[33] Vishal Srivastava, Priyam Tejaswin, Lucky Dhakad, Mohit Kumar, and Amar Dani. 2020. A Geocoding Framework Powered by Delivery Data. In *SIGSPATIAL'20.* ACM, 568–577.
[34] Qin TIAN, Yue GONG, Mengjun KANG, Shening MENG, and Qingyun DU. 2016. A comparative evaluation of online geocoding services in China. 41, 10 (2016), 1351–1358.
[35] Qin Tian, Fu Ren, Tao Hu, Jiangtao Liu, Ruichang Li, and Qingyun Du. 2016. Using an optimized Chinese address matching method to develop a geocoding service: a case study of Shenzhen, China. *ISPRS International Journal of Geo-Information* 5, 5 (2016), 65.
[36] Panrong Tong, Mingqian Li, Mo Li, Jianqiang Huang, and Xiansheng Hua. 2021. Large-Scale Vehicle Trajectory Reconstruction with Camera Sensing Network. In *MobiCom'21.* ACM, 188–200.
[37] Tin Vu, Solluna Liu, Renzhong Wang, and Kumarswamy Valegerepura. 2020. Noise Prediction for Geocoding Queries using Word Geospatial Embedding and Bidirectional LSTM. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems.* 127–130.
[38] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, and Gao Cong. 2021. A Survey on Trajectory Data Management, Analytics, and Learning. *ACM Comput. Surv.* 54, 2, Article 39 (March 2021), 36 pages.
[39] Zhengcong Yin, Andong Ma, and Daniel W Goldberg. 2019. A deep learning approach for rooftop geocoding. *Transactions in GIS* 23, 3 (2019), 495–514.
[40] Yukun Yuan, Desheng Zhang, Fei Miao, John A. Stankovic, Tian He, George Pappas, and Shan Lin. 2021. eRoute: Mobility-Driven Integration of Heterogeneous Urban Cyber-Physical Systems under Disruptive Events. *IEEE Transactions on Mobile Computing* (2021), 1–1.
[41] Hanyuan Zhang, Xingyu Zhang, Qize Jiang, Baihua Zheng, Zhenbang Sun, and Weiwei Sun. 2020. Trajectory similarity learning with auxiliary supervision and optimal matching. In *IJCAI'2020.* 11–17.
[42] Ping Zhang, Zhifeng Bao, Yuchen Li, Guoliang Li, Yipeng Zhang, and Zhiyong Peng. 2018. Trajectory-driven influential billboard placement. In *KDD'2018.* 2748–2757.
[43] Rui Zhang, Conrad Albrecht, Wei Zhang, Xiaodong Cui, Ulrich Finkler, David Kung, and Siyuan Lu. 2020. Map generation from large scale incomplete and inaccurate data labels. In *KDD'2020.* 2514–2522.
[44] Yan Zhao, Shuo Shang, Yu Wang, Bolong Zheng, Quoc Viet Hung Nguyen, and Kai Zheng. 2018. Rest: A reference-based framework for spatio-temporal trajectory compression. In *KDD'2018.* 2797–2806.
[45] Yu Zheng. 2015. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* 6, 3, Article 29 (May 2015), 41 pages.