SAInf: Stay Area Inference of Vehicles using Surveillance Camera Records

Zhipeng Ma[1,2,3]*, Chuishi Meng[2,3], Huimin Ren[2,3], Sijie Ruan[4], Jie Bao[2,3], Xiaoting Wang[2,3], Tianrui Li[1], Yu Zheng[2,3]
[1]Southwest Jiaotong University, Chengdu, China [2]JD iCity, JD Technology, Beijing, China
[3]JD Intelligent Cities Research, China [4]Beijing Institute of Technology, Beijing, China
mazhipeng1024@my.swjtu.edu.cn;{renhuimin5,mengchuishi1,baojie3,wangxiaoting35}@jd.com;sjruan@bit.edu.cn;
trli@swtjtu.edu.cn;msyuzheng@outlook.com

ABSTRACT

Stay area detection is one of the most important applications in trajectory data mining, which is helpful to understand human's behavior intentions. Traditional stay area detection methods are based on GPS data with relatively high sampling rate. However, because of privacy issues, accessing GPS data can be difficult in most real-world applications. Fortunately, traffic surveillance cameras have been widely deployed in urban area, and it provides us a novel way of acquiring vehicles' trajectories. All the vehicles that traverse by can be recognized and recorded in a passive way. However, the trajectory data collected in this way is extremely coarse, because the surveillance cameras are only deployed in important locations, such as crossroads. This coarse trajectory introduces two challenges for the stay area detection problem, i.e., whether and where the stay event occurs. In this paper, we design a two-stage method to solve the stay area detection problem with coarse trajectories. It first detects the stay event between a surveillance camera record pair, then uses a layer-by-layer stay area identification algorithm to infer the exact stay area. Extensive experiments based on real-world data were used to evaluate the performance of the proposed framework. Results demonstrate the proposed framework SAInf achieved a 58% performance improvement compared with SOTA methods.

KEYWORDS

Trajectory Data Mining, Stay Event Detection, Urban Computing

1 INTRODUCTION

Stay area detection is a common problem in trajectory data mining. Researchers use stay areas to understand the semantics of location and human mobility. However, traditional stay area detection algorithms are highly dependent on the GPS data with relatively high sampling rates [15, 21, 29, 30, 39]. Although GPS devices have been widely used to locate moving objects, it is still difficult for city managers to collect GPS data from all vehicles due to privacy issues [7, 11, 12] and budget limitations [4, 9]. In fact, the government only has access to GPS data for a small portion of vehicles which are under restrictive management, such as police cars, ambulances, and special trucks.

Fortunately, there are other types of sensors that can perceive vehicles' locations, such as traffic surveillance cameras, electronic toll collection systems, base stations, and etc. These sensing devices offer an alternative opportunity to solve the stay area detection problem. Among them, surveillance cameras are the most widely used, and more than one billion surveillance cameras have been deployed worldwide by 2021 [5]. Generally, when a vehicle traverses on the road, it can be captured by the surveillance cameras, and a record is generated with object detection and license plate recognition technology. Figure 1 shows a trajectory with a stay event, where the green line indicates the real trajectory of the vehicle, and the yellow dots indicate the camera positions. When the vehicle passes the camera in order, a record is generated which is indicated by the blue icons.

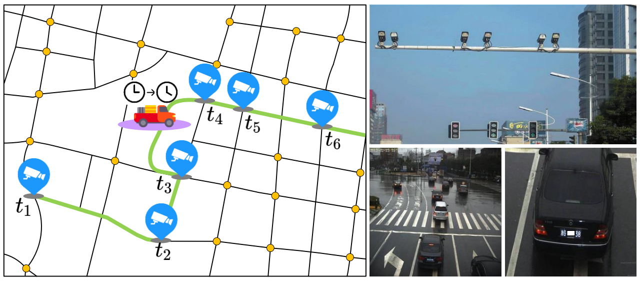


Figure 1: The Trajectory Recorded by the Surveillance Cameras.

Inferring the stay area has rich applications in real-world scenarios: 1) Discovering illegal activities of special vehicles. For special vehicles, such as chemical transport vehicles and construction waste vehicles, a stay event often indicates potential safety risks. The stay area of these vehicles may be concealed chemical storages, and the stay area of construction waste transfer vehicles relate to illegal dumping activities. 2) Discovering popular areas. Because surveillance cameras passively recognize all types of vehicles, we are able to discover popular areas of the city without sampling bias compared to the traditional method using GPS data collected from only one vehicle type, such as taxicabs [1, 2, 6]. 3) Mining vehicles' mobility. The stay behavior can reflect the travel intention of the vehicle. Although the semantic information of only one stay area

[*] The research was done when the first author was an intern at JD iCity, JD Technology, China and JD Intelligent Cities Research under the supervision of the second author.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

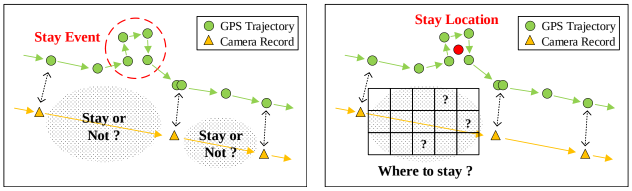
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599952

is limited, the pre- and post-sequences are able to help us better model the mobility of vehicles.

Compared to GPS devices, surveillance cameras have the following three advantages:

- **Wide coverage.** All moving vehicles and major areas in the city can be monitored by surveillance cameras. In a contrast, GPS trajectories can only be collected from a small portion of vehicles.
- **Privacy friendly.** Surveillance cameras are able to perceive mobility in the urban space without acquiring details of the object's trajectory.
- **Reliable access.** Compared to GPS devices, which are prone to irregular operation, and equipment failure, surveillance cameras are installed and maintained by city managers in a unified manner, allowing a stable collection of objects' visit records.

As shown in Figure 1, it is obvious that the sparse trajectory recognized by surveillance cameras is a down-sampling of the real trajectory of the vehicle, which presents a huge uncertainty for the stay area detection task. The uncertainty is mainly reflected in two aspects:



(a) Stay Event Uncertainty.

(b) Stay Area Uncertainty.

Figure 2: Challenges to Infer the Stay Areas.

- **Stay Event Uncertainty.** According to [37], a stay event is defined as the vehicle stay at a region beyond a certain temporal threshold. Traditional stay event detection algorithms can only extract cases which have a higher stay time than the sampling interval of devices. However, as shown in Figure 2(a), it is difficult to extract all available stay events (an available stay event is often set to stay longer than 5,10,15 min) from the surveillance camera records whose sampling interval is longer than 30min. Therefore, it is a challenge to detect whether a available stay event occurrs under a high record interval setting.
- **Stay Area Uncertainty.** Although the cameras have wide coverage, most of them are only deployed at key nodes of the road due to budget limitations. As a result, the stay event are usually not captured directly by the camera, i.e., keep stationary within the scene for a while. For example in Figure 2(b), We depict two consecutive camera records with stay events and estimate the potential stay areas. It is obvious that the further distance between two cameras, the larger candidate areas where the vehicle may stay. In our observation, there are more than 75% records whose potential stay areas exceed 10 square kilometers. Traditional stay area detection algorithms cannot be applied to such sparse trajectories.

To overcome the aforementioned challenges, we propose a framework for stay area detection, named SAInf (Stay Area Inference), which is a two-stage approach that models stay events and stay areas. In the first stage, the algorithm detects whether the stay events occur between two consecutive camera records, which is used to reduce the stay event uncertainty. In the second stage, we propose a layer-by-layer process combining coarse-grained and fine-grained selection to tackle the stay area uncertainty. On the one hand, the coarse-grained selection module uses the spatial distribution of stay events to generate the candidate region set. On the other hand, the fine-grained selection module infers the exact stay areas by calculating the stay probability of each candidate region.

Specifically, the proposed framework contains three main components: 1) data pre-processing, which detects the stay events from the GPS trajectories, and assigns the stay events to the corresponding camera record pairs. This component is only used in the offline learning phase for dataset build-up. 2) Stay event detection, which detects stay events within camera record pairs by testing travel speed. and 3) Stay area identification, which generates a candidate region set through modeling spatial distribution of stay events and selects the most likely stay area in its candidate region set.

The main contributions of the paper are summarized as follows:

- Our work is the first research to tackle the stay area detection problem with surveillance camera records. We analyze the stay area detection problem under a sparse trajectory setting and present the problem formulation and challenges in this paper.
- A novel stay area inference framework SAInf is proposed. SAInf provides a standard pipeline to address the uncertainties from camera records. In the mean time, we also design a model called StayNet for fine-grained stay area identification. StayNet takes in various factors, and models the complex relationship between candidate regions and stay events, .
- We evaluated the proposed method using a real-world dataset from Nantong, China. The results show that our method is efficient, with a performance improvement of about 58% compared to the current state-of-art baselines.

The rest of the paper is organized as follows: Section 2 describes the problem and the framework overview. Stay event detection is discussed in Section 3. Section 4 describes the detail of the stay area identification. Experiments and case study are given in Section 5. Related works are summarized in Section 6. Finally, we conclude in Section 7.

2 OVERVIEW

In this section, we first provide the preliminaries and used notations, then we define the stay area detection problem and outline our solution framework.

2.1 PRELIMINARY

Definition 1 (City Region). A city is divided into disjointed $M \times N$ grids based on latitude and longitude, where a grid denotes a region. All the grids form a region set $R = \{r_{11}, \ldots, r_{ij}, \ldots, r_{MN}\}$, where r_{ij} is the cell region in the i-th row and j-th column of the grid map.

Definition 2 (Surveillance Camera Record). A surveillance camera record is generated when a vehicle passes a camera. The record contains the mobility information, denoted as a 3-tuple $r =< vid, cid, t >$, where vid denotes vehicle ID, cid indicates the visited camera ID and t is the visit time. A camera is bounded to a fixed location $location(cid) =< lat_{cid}, lng_{cid} >$.

Definition 3 (Surveillance Camera Record Pair)**.** A Surveillance Camera Record (SCR) pair consists of two consecutive surveillance camera records. The pair is represented as $pr =< r_i, r_{i+1} >$, where r_i denotes a camera record.

Definition 4 (GPS Trajectory)**.** A trajectory is a sequence of GPS points, denoted as $tr =< p_1, p_2, \ldots, p_n >$, where each point $p_i =< lng_i, lat_i, t_i >$ indicates the longitude and latitude at a location time t_i. The points in the trajectory are organized chronologically.

Definition 5 (Stay Event)**.** A stay event se occurs when a moving object stays within a geographic region for a while, which is a triplet $se =< t_s, t_e, r_{ij} >$. t_s and t_e are the start and end timestamps of the stay event, and r_{ij} is a spatial grid.

Problem Statement. Given the surveillance camera record (SCR) pairs $PR = \{pr_i | i \in [1, \ldots, q]\}$ of vehicles, the stay area inference problem is to infer whether stay events occurred, and at which regions the vehicle stayed $\hat{R} = \{r_{ij} | i, j \in \text{stay region index}\}$.

2.2 Framework Overview

The framework of SAInf is elaborated in Figure 3, consisting of offline leanring and online inference phases. In the offline learning phase, the SAInf first constructs the dataset from collected surveillance camera records and GPS trajectories. Then, SAInf learns the relationship between SCR pairs and their corresponding stay events. In the online inference phase, the SAInf receives SCR pairs and uses the trained stay event detection module and stay area identification module to infer the stay event.

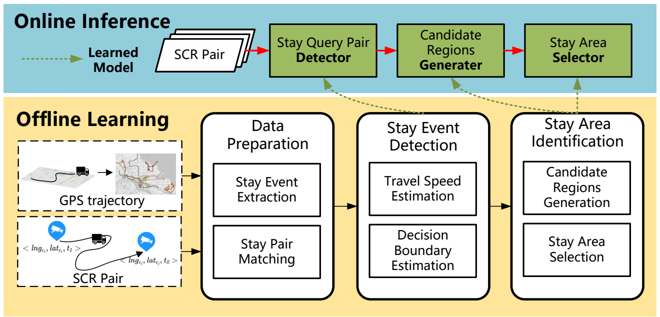


Figure 3: System Framework.

More specifically, there are three components:

Data Preparation. This component takes the SCR pairs and GPS trajectories to construct a dataset that is used in offline learning and contains two main tasks. 1) *Stay Event Extraction*,which performs trajectory noise filtering and stay area detection algorithms [37] to construct ground truth with GPS trajectory data. 2) *Stay Pair Matching*, which matches SCR pairs with stay events in chronological order.

Stay Event Detection. It provides a pipeline for detecting stay events from SCR pairs. Two main tasks are performed: 1) *SCR pair Modeling* , which characterizes the SCR pairs by modeling the travel speed. 2) *Stay Query Pair Detection*, which builds a statistical model to determine whether the stay event occurred within SCR pairs.

Stay Area Identification. This component takes in SCR pairs with stay events as input, and outputs the inferred stay area. Two steps are designed in this component: 1) *Candidate Regions Generation*, which generates a candidate region set of stay events by modeling the spatial properties of the SCR. 2) *Stay Area Selection*, which builds a deep learning model to estimate stay probability of each candidate region, and outputs the top-k regions with the highest probability.

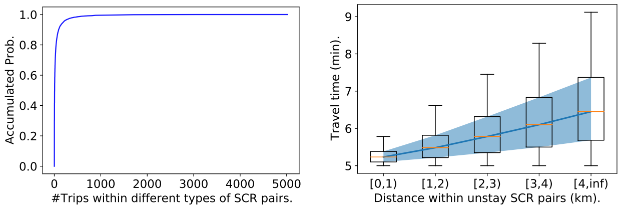
The data preparation process is an essential but straight-forward component, we will not elaborate more details in this paper. In the following sections, we will describe the stay event detection and stay area identification processes, respectively.

3 STAY EVENT DETECTION

In this module, we detect whether a stay event occurs in the SCR pair, which acts as inputs of the stay area identification.

Motivation. Common sense tells us tha the stay events are highly related to the travel time, since the vehicles spend more time in passing through the SCR pair with stay events. Ideally, we could learn a model for each camera pair if enough data is collected. However, the trips are distributed unevenly, in other words, most camera pairs are visited by very limited times. As shown in Figure 4(a), 95% camera pairs are visited by less than 100 times, and most of the camera pairs are visited less than 10 times. This indicates that most camera pairs do not have enough observations to build a separate detection model. In order to tackle this challenge, we aggregate the data and build a unified model.

For stay event detection, the premise of aggregation is to find a statistic to determine whether stay events occurred within an SCR pair. In general, the travel time between SCR pairs cannot be directly compared because the travel distances between SCR pairs are different. In order to eliminate the effect of route distance on travel time in individual SCR pairs, we adopt travel velocity instead of travel time to detect stay events, in order to build a unified statistic among different SCR pairs. However, travel velocity cannot be calculated directly because we only have information about the starting and ending records without the routes. In fact, the direct distance between the two cameras in the SCR pair is a lower bound of the route distance. Figure 4(b) also demonstrates the fact that the direct distance in the SCR pair shows the positive correlation with the travel time similar to route distance. Finally, we replace route distance by direct distance to estimate travel velocity of each SCR pairs.



(a) CDF of #Trips.

(b) Travel Time w.r.t Direct Distance.

Figure 4: Insights of Stay Event Detection.

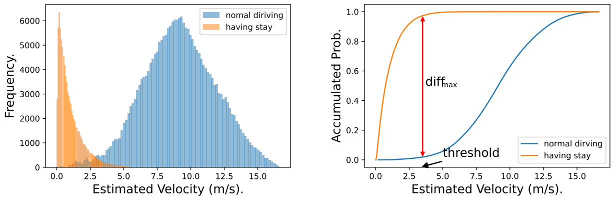
Inspired by the above insights, we first aggregate all SCR pairs to reduce the effect of skewed data distribution. Then, we use the estimated travel velocity as statistic v to determine the decision boundary. The equation is shown as eq. (1), which describes the statistic v for the SCR pair pr with two camera records r_i and r_{i+1}.

$$v = \frac{direct\_distance(r_i, r_{i+1})}{r_{i+1}.t - r_i.t} \tag{1}$$

Main Idea. We visualize the SCR pairs with stay events and those without stay events, and the result is shown in Figure 5(a). It indicates that stay events can be well separated from those without stay events using the estimated travel speed statistics v. Therefore, the key is to find a suitable threshold $\hat{v}$. Referring to the idea of KS test [10], we choose the position where the empirical cumulative probability function of the two statistics diviates the most. The formula is shown in eq. (2):

$$\hat{v} = \arg\max |CDF(V_{stay}) - CDF(V_{unstay})| \tag{2}$$

where V_{stay} and V_{unstay} are the population of travel speeds estimated from SCR pairs with stay events and SCR pairs without stay events, respectively. $CDF(\cdot)$ denotes the empirical CDF of the two population.



(a) Distribution of Velocity for Two Types of SCR Pairs.

(b) CDF of Velocity for Two Types of SCR Pairs.

Figure 5: Main Ideas of Stay Event Detection.

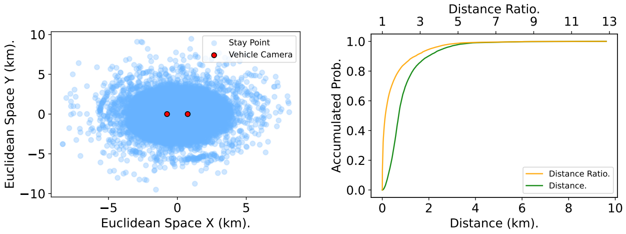
During the offline learning phase, the module takes SCR pairs with stay event D_{stay} and without stay event D_{unsaty}, and returns the velocity threshold $\hat{v}$. In the online inference phase, the module detects travel event based on travel speed v estimated from the SCR pair and threshold $\hat{v}$. Specifically, when the travel speed v is less than the threshold $\hat{v}$, the algorithm detects that stay events have occurred in the SCR pair, and pass the SCR pair as stay query pair to the following stay area detection task.

4 STAY AREA IDENTIFICATION

Considering the high uncertainty of the stay area identification problem, the proposed method needs to ensure both performance and computational efficiency. Inspired by the recommender system [23], we adopt coarse-grained and fine-grained selection in each SCR pair with potential stay events. This method balances the recall of candidate regions and the precision of stay area identification. More specifically, in the coarse-grained selection process, the algorithm determines the candidate region sets from the SCR pairs by modeling the reachable area. In the fine-grained selection stage, the algorithm selects the regions with high probability of stay event from the candidate region sets.

4.1 Candidate Regions Generation

Motivation. In this section, we model the spatial distribution of stay events using ellipse and design a statistic to generate candidate region set with respect to each SCR pair dynamically. This is inspired by two intuitions:



(a) Spatial distribution of stay events.

(b) CDF of two statistics.

Figure 6: Visualization at Stay Events within Stay Query Pairs.

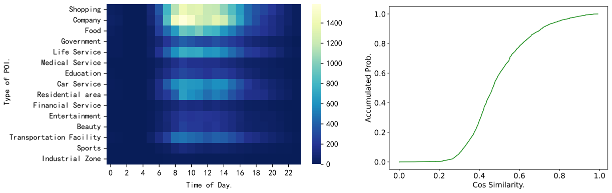
1) Given a starting point, an end point, and a travel time, the activity range of the vehicle is an ellipse with the starting and end points as focal points. We normalize the camera positions of stay query pairs (SCR pairs with stay events, returned by the last module) by rotation and transformation. As shown in Figure 6(a), red dots are normalized camera locations of SCR pair, and blue dots are ground truth stay events. It is obvious that the stay events are two-dimensional Gaussian distribution around the camera locations. As a result, we model the candidate area of a stay query pair as an ellipse. We denote the distance a vehicle traverses within each stay query pair as $2a$, and denote the direct distance between two cameras in the stay query pair as $2c$. Therefore, if a suitable distance $2a$ is found, we can generate the candidate region set of the stay query pairs based on conic section and predefined grids.

2) The candidate regions need to be adjusted dynamically according to specific SCR pairs. As the green line shown in Figure 6(b), distance between SCR pairs ranges from meters to several kilometers. In other words, we should have less candidate stay regions when SCR distance is short, and have more candidate stay regions when SCR distance is long. Based on the properties of ellipses, we devise a new statistic, distance ratio τ, which refers to $\frac{a}{c}$ and dynamically adjusts the candidate regions by $2c$. The yellow line in Figure 6(b) shows that distance ratio τ has the similar statistical properties as distance and can vary with the direct distance within each stay query pair.

Main Idea. Based on the previous discussion, we choose to use the distance rate as the statistic to estimate dynamically the parameters of the elliptic equation and generate a region set based on ellipses and predefined grids. The threshold τ was calculated based on the distance rate of the 95th percentile. Because Figure 6(a) shows some outliers, and they are mostly noise generated by camera false alerts and missed detection. A threshold τ with outliers will put more grids in the candidate region set and affect the performance of stay region selection.

4.2 Stay Area Selection

According to the previous process, the candidate region generation module estimates thresholds to generate the candidate region set for each stay query pair. In the stay area selection module, our task is to select k regions from the candidate region set. However, due to the complexity of the stay pattern, three challenges arise. The first challenge is how to extract the relationship between regions and stay events. Second, encoding the spatio-temporal context of the stay query pair is another challenge considering the various spatio-temporal correlations. Finally, how to effectively fuse these information to calculate the stay probability of each region from the candidate region set is another challenge.



(a) Stay Event Heatmap.

(b) Differences between regions in the candidate set.

Figure 7: Insight of Stay Area Selection.

Main Idea. To solve these problems, we design a model StayNet, which outputs the stay probability of each region in the candidate region set. It is designed based on the following three insights: 1) Stay events are affected by spatio-temporal context. Figure 7(a) presents the heat map of the vehicle stayed at various POIs and periods. It is obvious that vehicles tend to stay in similar types of locations at similar periods. For example, hazardous chemical vehicles need to transport chemicals, they often stay near companies, stores, medical and other facilities centrally. And taxis inclines to visit hot urban areas frequently. We propose to capture the spatio-temporal information of the stay query pairs through time periods, visited surveillance camera, weather, and etc. 2) Furthermore, stay behavior is intentional and is relevant to the region semantically. Relying on the fact that semantic information of the region is described by the contained POI, we use POI to model regions to further capture the relationship between stay events and regions. 3) Figure 7(b) shows the average cosine similarity within the set for each candidate region set. It reveals that the differentiation of the regions in the set is very low because the average cosine similarity in more than half of the sets exceeds 0.5. Therefore, it is very challenging to select the region containing the stay events from the candidate region set. Here we use the powerful information interaction capabilities in the transformer to help extract the differences between regions. Overall, the model calculates the stay probability of each region in the candidate region set and returns the k region with the highest probability.

Model Overview. In practice, we designed a stay area selection model, called StayNet. It selects the top-k stay regions with maximum probability for each stay query pair. Figure 8 depicts the structure of StayNet, which consists of three components:

- **Candidate Region Representation**, which models all regions in the candidate region set uniformly using TF-IDF vector of POIs, and output variable representation of each region;
- **Spatio-temporal Context Encoding**, which extracts and embeds spatio-temporal information of each stay query pair, and combines the embeddings to generate a hidden representation that indicates context of stay;
- **Knowledge Fusion**, which fuses and further enhances each region representation with the spatio-temporal context representation in an adaptive manner so that an ideal prediction result can be achieved.

The pipeline of our proposed model StayNet is as follows: StayNet takes stay query pairs and their corresponding candidate regions as input. Firstly, the candidate region representation component takes the candidate region set and produces a representation of each region as the key. Then, the spatio-temporal context encoding component encodes the time, location, and other information of the stay query pairs to the spatio-temporal context vectors. The spatio-temporal context vectors are used as the queries and values. Finally, the queries, keys, and values are fed to the knowledge fusion component. For each region, the knowledge fusion component fuses this information based on the attention mechanism and returns stay probability.

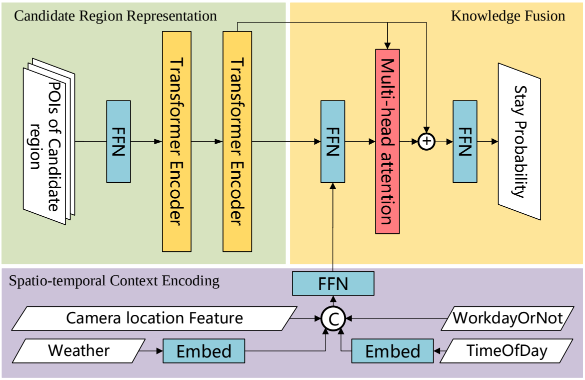


Figure 8: The Structure of StayNet.

Candidate Region Representation. The functional properties of the location is very helpful to infer the stay event. A better distinguish between expressive features and inexpressive features can help us detect stay events more accurately. As a result, we choose a transformer to model the variable-length candidate region set.

Transformer [27] has been widely used in natural language processing, computer vision, and sequence modeling, which follows the architecture of the encoder-decoder structure. Both the encoder and decoder are composed of stack identical Transformer Block (TB) layers. Each transformer block has two sub-layers, including a multi-head self-attention mechanism (MultiHead(·)), and a simple position-wise fully connected feed-forward network (FFN(·)). The residual connection [13] is used around each of the two sub-layers, followed by a layer normalization [3].

The MultiHead(·) is an ensemble of h single-head attention, which concate all head computed in parallel and project them once again. A single-head attention mechanism is described in formula eq. (3), which calculates the weighted sum of the values, where the weights are obtained by interacting with the corresponding query and key using the softmax function:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{3}$$

where $\frac{1}{\sqrt{d_k}}$ is the scaling factor, $Q, K \in \mathbb{R}^{L\times d_{model}}$ is the query matrix and the key matrix,$V \in \mathbb{R}^{L\times d_{model}}$ is the key matrix which from multiple queries, keys and values packed together, respectively. L is the collection length of queries, keys, and values. Further, the MultiHead(·) is as follows:

$$\begin{aligned} \text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= [\text{head}_1;\ldots;\text{head}_h]\mathbf{W}^O \\ \text{where head}_i &= \text{Attention}(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V) \end{aligned} \tag{4}$$

where $W_i^Q \in \mathbb{R}^{d_{model}\times d_k}$,$W_i^K \in \mathbb{R}^{d_{model}\times d_k}$, $W_i^V \in \mathbb{R}^{d_{model}\times d_v}$, $W^O \in \mathbb{R}^{hd_v\times d_{model}}$ are the learnable parameter matrices.

The FFN(·) layer contains a fully connected feed-forward network, which is applied to each position separately and identically. It consists of two direct transformations with a ReLU activation with learnable parameters $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$:

$$\text{FFN}(\mathbf{X}) = \max(\mathbf{0}, \mathbf{XW}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{5}$$

Consider a set $X \in \mathbb{R}^{L\times d}$ with L elements and each element with d-dimensional features, the process of passing through Transformer Encoder with multiple Transformer Block layers is as follows:

$$\begin{aligned} \mathbf{X}^0 &= \mathrm{X} \\ \tilde{\mathbf{X}}^{j+1} &= \text{LayerNorm}(\text{MultiHead}(\mathbf{X}^j,\mathbf{X}^j,\mathbf{X}^j) + \mathbf{X}^j) \\ \mathbf{X}^{j+1} &= \text{LayerNorm}(\text{FFN}(\tilde{\mathbf{X}}^{j+1}) + \tilde{\mathbf{X}}^{j+1}), j = 1,\ldots,m-1 \end{aligned} \tag{6}$$

where m is the number of transformer block layers. The X^m is the final output of the transformer block. Note that elements in the set are disordered so that positional encoding layer is removed.

In this component, we use the vector composed of TF-IDF values for 15 types of POI and the distance from region to two cameras to model each region, and combine all regions in a candidate region set as the input X_r, $X_r \in \mathbb{R}^{L\times 15}$. It is expected that the differential representations between regions are captured through double transformer block layers as encoders after processing with a feed-forward network:

$$\begin{aligned} \tilde{X_r} &= \text{TransEnc}(\text{FFN}(X_r)) \\ E_r &= \text{TransEnc}(\tilde{X_r}) \end{aligned} \tag{7}$$

Spatio-temporal Context Encoding. Because the stay event is also related to its own spatio-temporal context, we extract three types of features in the stay query pair to represent the context information, including spatial features, temporal features, and weather features. The spatial features x_s are composed of the latitude, longitude and location embedding of the two camera positions visited, and the corresponding grid coordinates. Camera position embedding is pre-trained on camera check-in trajectory by DeepWalk [22]. There are two features that belong to the temporal type. One is the index of the discrete-time bin x_b corresponding to the start and end time of the stay query pair, the other is the binary value x_d indicating whether the time slot is on workdays or weekends. Finally, feature x_w is the weather type during the occurrence in each stay query pair. During processing, the x_d and x_w are fed firstly into an embedding layer to obtain the dense representation. Then, we concatenate them together and send them to a feed-forward network to obtain the stay condition representation e_{st}. Details are shown in eq. (8):

$$\begin{aligned} e_w &= \text{Embed}(x_w) \\ e_b &= \text{Embed}(x_b) \\ e_{st} &= \text{FFN}([x_s; e_w; e_b; x_d]) \end{aligned} \tag{8}$$

Knowledge Fusion. Knowledge fusion component takes candidate region representation E_r and stay condition representation e_{st} as the input, enhances E_r using cross attention based on e_{st} and returns stay probability of each candidate region. Moreover, residual connections are used to ensure the stability of the training. Knowledge fusion is defined as follows:

$$\begin{aligned} E_{st} &= \text{MultiHead}(e_{st}, E_r, e_{st}) + E_r \\ \hat{y} &= \text{FFN}(E_{st}) \end{aligned} \tag{9}$$

In the training phase, we use BCEloss to optimize the StayNet:

$$loss = -\frac{1}{n}\sum_i^n w_i\left[y_i \log(\hat{y_i}) + (1-y_i)\log(1-\hat{y_i})\right] \tag{10}$$

where $\hat{y}$ is predicted result, y is corresponding ground truth. It should be noted that the imbalance of positive and negative samples in the candidate region set. So, we use the negative to positive sample ratio w to weight the positive samples in BCEloss [35].

5 EXPERIMENTS

5.1 Experimental Settings

We use a real world dataset which contains surveillance camera records and GPS trajectory data for 2,182 vehicles that were collected from Nantong, China from March 1, 2021 to March 18, 2022.

- **Surveillance Camera Record.** A total of 14,058 smart surveillance cameras are deployed at intersections in Nantong. When a vehicle passes them, a log record containing timestamp, longitude, latitude, license plate number and the corresponding camera ID is generated. The dataset contains 1.22 million records generated by 2182 vehicles in 175 days.
- **GPS Trajectory.** They are raw GPS logs generated by the vehicle's navigation device, each record contains license plate number, longitude, latitude and timestamp. In the paper, 230 million data for 265 days from 2182 vehicles were used. These data are pre-processed and produce stay events as ground truth.
- **POI.** We collected 15 categories of POI data from Nantong for characterizing the area using services.
- **Weather.** Weather data is collected from open API services to model the context of stay event.

In practice, after the GPS trajectory and camera check-in records are processed by the data pre-processing module, the stay events detected from the GPS trajectory and the camera check-in records are combined in chronological order to obtain two types of SCRs. The POI data is projected onto the predefind grids and the tf-idf vectors of regions are calculated. Last, a total of 329.3 thousand SCR pairs are generated, of which first 60% are used for training, middle 20% for validation, and finally 20% for testing in chronological order.

Evaluation Metrics. We choose hit ratio, which measures how many real stay regions can recall successfully from its k query results. It is formulated as

$$hit@k = \frac{\#\text{hits of topK}}{\#\text{stay region}} \tag{11}$$

Hyperparameters. The stay area detection algorithm uses the two thresholds are 50 meters and 300 seconds. The region size is set to 1km × 1km. The 2-layer Transfomer Encoder is used in StayNet. 4 and 8 headers are used in the transformer and knowledge fusion, respectively. The hidden size of FC layer are all set to 128. Weather type and time of day index is embedded to $\mathbb{R}^2$. In training phase, we adopt Adam optimizer with β_1= 0.9 and β_2= 0.999 and use an initial learning rate 7e-3. The training epoch is set to 100 rounds.

5.2 Comparison of Frameworks

To the best of our knowledge, there is no solution that can be used to solve the problem well. Therefore, we designed four baselines based on the enforcement experience of city managers.

- **Random Infer (RInf).** For each SCR pair, we do not detect stay events, and randomly select k regions in the 5km circles centered at the midpoint of the each SCR pair.
- **Spatial Heuristic Infer (SHInf).** Without detecting stay events, the pipeline determines the candidate range based on the empirical distance (5km) and performs spatial heuristic selection in regions within the range. Spatial heuristic selection sorts the regions in descending order by frequency of stay events, and returns the top-k regions.
- **Velocity Heuristic Infer (VHInf).** The empirical velocity (3m/s) is used as the threshold for detecting stay events. We use the empirical velocity and the interval of the SCR pair to calculate the candidate range and perform spatial heuristic selection on the regions within the range.
- **Spatial-Velocity Heuristic Infer (SVHInf).** A hybrid of SHInf and VHInf. Pipeline first uses empirical velocity (3m/s) to detect stay events, then uses empirical distance (5km) to determine candidate ranges.

We also compare SAInf with its three variants:

- **SAInf-nD.** This variant uses the empirical velocity as a threshold to detect stay events. Settings of stay area inference are consistent with SAInf.
- **SAInf-nS.** This variant uses fixed empirical spatial threshold to determine the range of candidate region set instead of adaptive stay region generation.
- **SAInf-nC.** Instead of using StayNet as the selection model, this variant uses spatial heuristic selection.

Table 1: Stay Area Infer Evaluation.

Methods	hit@1	hit@3	hit@5	Total
RInf	0.0202	0.0627	0.1019	0.1848
SHInf	0.1966	0.4023	0.5055	1.1044
VHInf	0.2141*	0.4155*	0.5218*	1.1514*
VSHInf	0.1960	0.3856	0.4828	1.0644
SAInf-nD	0.4170	0.6430	0.7263	1.7863
SAInf-nC	0.4084	0.6224	0.7140	1.7448
SAInf-nS	0.2785	0.4839	0.5806	1.3430
SAInf(Ours)	**0.4373**	**0.6659**	**0.7477**	**1.8509**

Evaluation. Table 1 shows the performance of the proposed framework as compared to all other competing framework. Our proposed SAInf significantly outperforms all competing baselines by achieving the highest all metric. Overall 58.03% performance improvement in SAInf compared to the current best method. And, SAInf in hit@1hit@3 and hit@5 improved by 95.56%, 57.06% and 43.41% respectively. RInf and SHInf do not detect stay events, but perform stay region selection for all input SCR pairs. This rough processing results in part stay SCR pairs being lost, which greatly affects the subsequent stay region inference. And even worse , the large and invalid results generated a waste of computational and storage resources. VHInf and VSHInf perform stay event detection on SCR pairs by empirical speed to avoid losting SCR pair with stay events. In contrast, the candidate region set for VHInf depends on the empirical velocity and the duration of the SCR pair, while the candidate region set for VSHInf is specified by the empirical distance. It suggests that candidate region set that vary dynamically with the SCR pair are more helpful in the stay region selection. Among all other baselines and variants, VHInf obtains the best performance.

For the three variants of SAInf, the performance improvement of SAInf-nS is the most significant, which demonstrates the effectiveness of StayNet. Moreover, it can be found that the performance of both SAInf-nC and SAInf-nV using empirical parameters is lower than our framework, which indicates the superiority of our proposed stay event detection algorithm and candidate region generation algorithm. Specifically, in the candidate region generation, the recall of the algorithm and the number of candidate region are trade-off. In general, to recall all possible stay regions would increase the number of candidate regions, resulting in a poor selection phase. When the sides of the region are 1000m, our candidate region generation algorithm can guarantee the average number of candidate regions to be 32 under the condition that 95% of the results are recalled, but the algorithm based on empirical parameters makes the average number of candidate regions rise to 101 under the condition that 95% of the results are recalled. The experimental results prove that we have reached a better balance.

Parameter Sensitivity. We conducted experiments in a square region with side lengths of 500m, 1000m and 1500m respectively, and the results of the experiments are presented in Figure 9 which shows the best baseline and our method and its variants. Overall, all variants showed improvement compared to the baseline method, and the most significant improvement was seen at the 1500m setting. The performance improvement becomes larger as the side length of the region becomes larger. This is similar to our intuition that the larger the granularity of the region the better the performance of the model, but the more difficult it is to apply in practice because of the large scope of the region. To balance the performance and

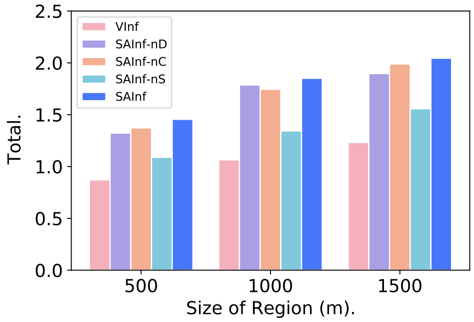


Figure 9: Experimental Results of Different Framework.

management, we choose a region with a side length of 1000m as the minimum unit when it is deployed.

5.3 Comparison of Selection Model

The baselines for selection model falls into three categories, which are heuristics, instance modeling, and joint modeling. The heuristic method contains both spatial heuristic selection (SHS) and spatio-temporal heuristic selection (STHS). The spatial heuristic selection returns the k regions with the highest frequency in the candidate region set. The spatio-temporal heuristic selection takes into account the temporal dimension which is start time of the SCR pair. The instance modeling approach models each region in the candidate region set individually as a binary classification problem, and returns the k regions with the highest stay probability. Such methods include logistic regression (LR), eXtreme Gradient Boosting (XGBOOST) and multilayer perceptron (MLP). The joint modeling combines all regions from the candidate region set together into the model, and return the stay probabilities in each region to increase the information interaction among regions. RNN and Transformer both belong to the this type.

Two variants are listed as follows:

- **SatyNet-nFus.** Attention mechanism is replaced by addition in SatyNet-nFus.
- **SatyNet-nGlo.** We remove the spatio-temporal context encoding and use only the output of the vehicle representation for knowledge fusion.

Table 2: Area Selection Evaluation.

Methods	**hit@1**	**hit@3**	**hit@5**	**Total**
Random	0.1063	0.2520	0.3426	0.7009
SHS	0.3303	0.5740	0.6887	1.5930
STHS	0.3394	0.5614	0.6596	1.5604
LR	0.2977	0.5999	0.7526	1.6502
XGBOOST	0.4737*	0.7191	0.8314	2.0242
MLP	0.4182	0.7107	0.8232	1.9521
RNN	0.3285	0.6124	0.7536	1.6945
Transformer	0.4663	0.7434*	0.8487*	2.0584*
StayNet-nGlo	0.4777	0.7559	0.8598	2.0935
StayNet-nFus	0.5149	0.7839	0.8856	2.1843
StayNet (Ours)	**0.5187**	**0.7899**	**0.8869**	**2.1954**

Evaluation. The results of area selection are presented in Table 2. It can be seen that compared to the current best algorithm, proposed StayNet has an overall improvement of 6.7% and in hit@1,hit@3 and hit@5 improved by 11.2%, 6.3% and 4.5%, respectively. Specifically, with the exception of StayNet and its variants, XGBOOST achieved the highest hit@1, demonstrating the superiority of instance modeling. While transformer achieved the highest hit@3,hit@5 and Total result. This may be related to the fact that the joint modeling approach is able to better capture the functional differences among regions within the candidate region set, and such differences describe more essentially the interaction of the functionality of regions on the stay behavior.

From the different types of methods, all heuristics show a significant improvement compared to random selection, which indicates that the functionality and spatio-temporal context of region has a strong correlation with the stay behavior. It is worth noting that although STHS has an improvement in hit@1 compared to SHS, it then decreases in overall performance. Only STHS takes into account the effect of temporal dimension, but due to the limitation of observation data, STHS cannot accurately model the effect of regions on stay behavior. The instance modeling approach can make more efficient use of historical observation data, and achieves better results than heuristics by modeling the nonlinear correlation between regions and the stay behavior pattern. It is important that in the instance modeling approach, each region contains a POI tf-idf vector of the region and spatio-temporal context information. However, due to the fact that the instance modeling uses individual view of each region in the candidate region set, it cannot accurately quantify the differences among regions and leads to a lower differentiation in stay probabilities. MLP and XGBOOS have similar results in hit@3 and hit@5, and XGBOOST achieves the best results for hit@1 except for our method. It indicates that XGBOOST can effectively attenuate the effect of useless features in the region on the results, because of its powerful feature filtering ability. In fact, this is similar in implication to the purpose of joint modeling, which is to better differentiate features among regions. In contrast, the joint modeling approach can uniformly observe all instances in the candidate region set and effectively improve the accuracy of ranking. Among them, transformer achieves competitive results in most metrics. RNN degrades the performance because the sequential modeling approach cannot process the instances in the stay region set in parallel.

Comparing with variants of our proposed method, we found that the biggest improvement was achieved by encoding spatio-temporal contextual information separately and then performing knowledge fusion. From StayNet-nFus to StayNet, the knowledge fusion component of StayNet using the attention mechanism is able to better fuse spatial-temporal context encoding and region representation.

Parameter Sensitivity. In addition, we also evaluated the performance of different selection models under the three region settings. The best baseline, our model and its variants are compared in Figure 10. Our method and its variants achieve the maximum boost on a region with a 500m side length, comparing to previous discussions on frameworks which improves the most with 1500m side length. This is related to the number of candidate regions and the distribution of POIs; when the number of candidate regions

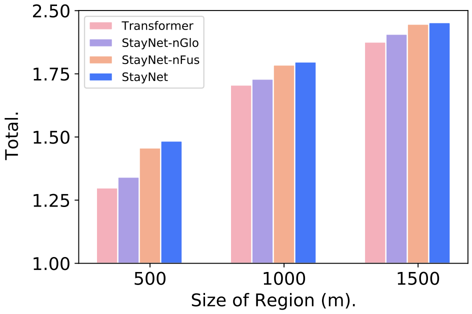


Figure 10: Experimental Results of Different Selection Model.

becomes larger, the task becomes more difficult and the model performance decreases. Since POIs are spatially clustered, when the size of the region becomes small, the POI loses its ability to characterize the regions lacking significant POI. At the same time, as the regions become larger, the number of candidate regions becomes smaller, the difficulty of the task decreases, and the performance improvement of our model diminishes.

5.4 Case Study

We further give a case study to test the effectiveness of SAInf in a real world setting. We use SAInf to infer the potential stay areas of the vehicle in real time and compare the results with the stay events of the corresponding vehicle equipped with a GPS device. It was found that 3 of the top 5 regions output by SAInf had vehicles actually staying. Based on the results of the SAInf, government managers can develop the order of inspection to discover illegal construction waste dumping sites.

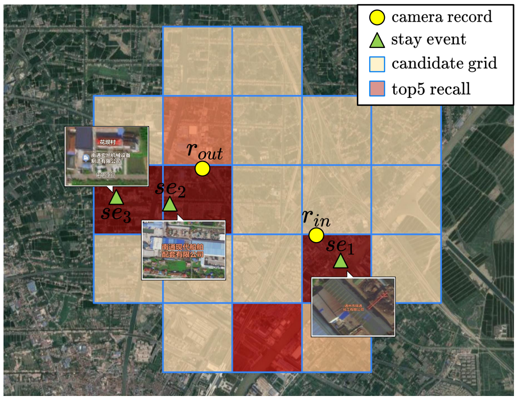


Figure 11: Case Study.

6 RELATED WORK

Stay Event Mining. The stay of mobile objects is often accompanied by rich semantics. Mining stay events can help us understand the mobility of objects and the functionality of location. In the past decades, with the popularity of GPS devices, a large number of stay event mining research efforts have emerged. It is divided into three main areas of work. 1) Event Discovery. [25, 33, 40] associated clusters of stay events with specific behaviors to mine stay events with semantics. 2) Location Discovery. [15, 21, 29] use clustering algorithm based on stay events to discover semantic regions from GPS trajectory data. [17, 39] further exploring the relations between the regions to predict the next region and provide location-based services. 3) Mobility Understanding. [14, 20] modeled the user's location history and analyzed their daily behavior pattern based on the sequences of historical stay events. However, these works are designed based on GPS trajectory data. Limited by the privacy and coverage of GPS device, we can only observe with sampling bias. In contrast to these works, we use surveillance camera records to capture vehicles' stay patterns and provides the underlying support for subsequent semantic mining.

Surveillance Camera Records Mining. Surveillance camera records mining refers to the use of camera-captured vehicle records to discover various knowledge. Surveillance camera records are widely used to capture the traffic status of the whole urban [8, 34]. And, some work has emerged to effectively improve the quality of data to better support downstream applications [19, 26, 32]. Chen et al. [8] used OD pairs which extracted from camera records to detect potential community for each vehicle. However these works tend to address the problem of unbalanced data distribution and missing data due to the instability of the devices. In our work, we focus on solving the stay uncertainty in surveillance camera record mining. Although Chen [8] attempt to solve the problem, their approach has strong assumptions and is not applicable to our scenario.

Urban Computing. Urban computing [38] is an interdisciplinary field that involves the research and application of computing technologies in urban areas. urbanization, such as crowd flows prediction [18, 36], air quality prediction [28, 31], and resource scheduling [16, 24]. In our work, we propose a framework to infer stay areas of vehicles using surveillance camera records, which can help city managers improve law enforcement efficiency

7 CONCLUSION

In this paper, we explore the stay area detection problem with camera records for the first time and design a framework SAInf to solve it. SAInf models the effect of specific spatio-temporal context on stay behavior by learning the relationship between SCR pairs and stay events. In practice, SAInf detects stay events through a two-stage design. It first discovers the SCR pair containing stay events through the stay event detection module, then infers the locations of the stay events with the stay area identification module. Experiments show SAInf outperforms baselines by 58% on a real-world dataset. In the future, we will further explore the design of other modules in the framework to achieve end-to-end framework. One direction is obtaining a more robust potential stay candidate set by predicting the stay interval time within SCR pairs to improve the performance.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2019YFB2103201), Beijing Nova Program (Z211100002121112) and National Natural Science Foundation of China (62176221). Sijie Ruan is supported by Beijing Institute of Technology Research Fund Program for Young Scholars (6120220113).

REFERENCES

[1] Antonio Luca Alfeo, Mario GCA Cimino, Sara Egidi, Bruno Lepri, and Gigliola Vaglini. 2018. A stigmergy-based analysis of city hotspots to discover trends and anomalies in urban transportation usage. *IEEE Transactions on Intelligent Transportation Systems* 19, 7 (2018), 2258–2267.

[2] Antonio Luca Alfeo, Mario Giovanni CA Cimino, Sara Egidi, Bruno Lepri, Alex Pentland, and Gigliola Vaglini. 2017. Stigmergy-based modeling to discover urban activity patterns from positioning data. In *Social, Cultural, and Behavioral Modeling: 10th International Conference, SBP-BRiMS 2017, Washington, DC, USA, July 5-8, 2017, Proceedings 10*. Springer, 292–301.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[4] Jie Bao, Ruiyuan Li, Xiuwen Yi, and Yu Zheng. 2016. Managing massive trajectories on the cloud. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–10.

[5] P Bischoff. [n. d.]. Surveillance camera statistics: which city has the most CCTV cameras? May 2021.

[6] Li Cai, Haoyu Wang, Cong Sha, Fang Jiang, Yihang Zhang, and Wei Zhou. 2021. The mining of urban hotspots based on multi-source location data fusion. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[7] Sujin Cai, Xin Lyu, Xin Li, Duohan Ban, and Tao Zeng. 2021. A Trajectory Released Scheme for the Internet of Vehicles Based on Differential Privacy. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2021), 16534–16547.

[8] Kai Chen, Yanwei Yu, Peng Song, Xianfeng Tang, Lei Cao, and Xiangrong Tong. 2020. Find you if you drive: Inferring home locations for vehicles with surveillance camera data. *Knowledge-Based Systems* 196 (2020), 105766.

[9] Philippe Cudre-Mauroux, Eugene Wu, and Samuel Madden. 2010. Trajstore: An adaptive storage system for very large trajectory data sets. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*. IEEE, 109–120.

[10] Zvi Drezner, Ofir Turel, and Dawit Zerom. 2010. A modified Kolmogorov–Smirnov test for normality. *Communications in Statistics—Simulation and Computation®* 39, 4 (2010), 693–704.

[11] Sheng Gao, Jianfeng Ma, Weisong Shi, Guoxing Zhan, and Cong Sun. 2013. TrPF: A trajectory privacy-preserving framework for participatory sensing. *IEEE Transactions on Information Forensics and Security* 8, 6 (2013), 874–887.

[12] Soheila Ghane, Lars Kulik, and Kotagiri Ramamohanarao. 2019. TGM: A generative mechanism for publishing trajectories with differential privacy. *IEEE Internet of Things Journal* 7, 4 (2019), 2611–2621.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[14] Yang Ji, Chunhong Zhang, Zhihao Zuo, and Jing Chang. 2012. Mining user daily behavior based on location history. In *2012 IEEE 14th International Conference on Communication Technology*. IEEE, 881–886.

[15] Sonia Khetarpaul, Rashmi Chauhan, SK Gupta, L Venkata Subramaniam, and Ullas Nambiar. 2011. Mining GPS data to determine interesting locations. In *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*. 1–6.

[16] Yexin Li, Yu Zheng, and Qiang Yang. 2018. Dynamic bike reposition: A spatiotemporal reinforcement learning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1724–1733.

[17] Yung-Hsiang Lin, Chien-Hsiang Lai, and Po-Ruey Lei. 2014. Mining top-k relevant stay regions from historical trajectories. In *Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2014 International Workshops: DANTH, BDM, MobiSocial, BigEC, CloudSD, MSMV-MBI, SDA, DMDA-Health, ALSIP, SocNet, DMBIH, BigPMA, Tainan, Taiwan, May 13-16, 2014. Revised Selected Papers 18*. Springer, 293–304.

[18] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. 2019. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 1020–1027.

[19] Zongyu Lin, Guozhen Zhang, Zhiqun He, Jie Feng, Wei Wu, and Yong Li. 2021. Vehicle Trajectory Recovery on Road Network Based on Traffic Camera Video Data. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 389–398.

[20] Hongwei Liu, Gang Wu, and Guoren Wang. 2014. Tell me where to go and what to do next, but do not bother me. In *Proceedings of the 8th ACM Conference on Recommender systems*. 375–376.

[21] Chun-Ta Lu, Po-Ruey Lei, Wen-Chih Peng, and Ing-Jiunn Su. 2011. A framework of mining semantic regions from trajectories. In *International Conference on Database Systems for Advanced Applications*. Springer, 193–207.

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[23] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.

[24] Sijie Ruan, Jie Bao, Yuxuan Liang, Ruiyuan Li, Tianfu He, Chuishi Meng, Yanhua Li, Yingcai Wu, and Yu Zheng. 2020. Dynamic public resource allocation based on human mobility prediction. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 4, 1 (2020), 1–22.

[25] Sijie Ruan, Zi Xiong, Cheng Long, Yiheng Chen, Jie Bao, Tianfu He, Ruiyuan Li, Shengnan Wu, Zhongyuan Jiang, and Yu Zheng. 2020. Doing in One Go: Delivery Time Inference Based on Couriers' Trajectories. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2813–2821.

[26] Panrong Tong, Mingqian Li, Mo Li, Jianqiang Huang, and Xiansheng Hua. 2021. Large-scale vehicle trajectory reconstruction with camera sensing network. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 188–200.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[28] Junshan Wang and Guojie Song. 2018. A deep spatial-temporal ensemble model for air quality prediction. *Neurocomputing* 314 (2018), 198–206.

[29] Xiaohan Wang, Zepei Zhang, and Yonglong Luo. 2022. Clustering Methods Based on Stay Points and Grid Density for Hotspot Detection. *ISPRS International Journal of Geo-Information* 11, 3 (2022), 190.

[30] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. 2010. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. 442–445.

[31] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2018. Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 965–973.

[32] Fudan Yu, Wenxuan Ao, Huan Yan, Guozhen Zhang, Wei Wu, and Yong Li. 2022. Spatio-Temporal Vehicle Trajectory Recovery on Road Network Based on Traffic Camera Video Data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4413–4421.

[33] Qingying Yu, Yonglong Luo, Chuanming Chen, and Xiaoyao Zheng. 2019. Road congestion detection based on trajectory stay-place clustering. *ISPRS International Journal of Geo-Information* 8, 6 (2019), 264.

[34] Yanwei Yu, Xianfeng Tang, Huaxiu Yao, Xiuwen Yi, and Zhenhui Li. 2019. Citywide traffic volume inference with surveillance camera records. *IEEE Transactions on Big Data* 7, 6 (2019), 900–912.

[35] Cheng Zhang and Chen Li. 2021. Neural Collaborative Filtering Recommendation Algorithm Based on Popularity Feature. In *2021 International Conference on Culture-oriented Science & Technology (ICCST)*. IEEE, 316–323.

[36] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*.

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

[38] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 1–55.

[39] Yu Zheng and Xing Xie. 2010. Learning location correlation from gps trajectories. In *2010 Eleventh International Conference on Mobile Data Management*. IEEE, 27–32.

[40] Zheng Zhu, Huimin Ren, Sijie Ruan, Boyang Han, Jie Bao, Ruiyuan Li, Yanhua Li, and Yu Zheng. 2021. Icfinder: A ubiquitous approach to detecting illegal hazardous chemical facilities with truck trajectories. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 37–40.