Multi-View Context Awareness based Transport Stay Hotspot Recognization

Tao Wu
East China Normal University
Jiali Mao
jlmao@dase.ecnu.edu.cn

East China Normal University
Yifan Zhu
East China Normal University
Kaixuan Zhu
East China Normal University
Aoying Zhou
East China Normal University

Research Article

Keywords: transport stay hotspot, multi-view clustering, semantic contextual feature, ensemble learning

Posted Date: January 10th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3834127/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at World Wide Web on July 18th, 2024. See the published version at https://doi.org/10.1007/s11280-024-01256-5.

Multi-View Context Awareness based Transport Stay Hotspot Recognization

Tao Wu[1], Jiali Mao[1*], Yifan Zhu[1], Kaixuan Zhu[1], Aoying Zhou[1]

[3]School of Data Science and Engineering, East China Normal University, Shanghai, 200062, China.

*Corresponding author(s). E-mail(s): jlmao@dase.ecnu.edu.cn;
Contributing authors: 52195100007@stu.ecnu.edu.cn;
51265903116@dase.ecnu.edu.cn; 51215903072@dase.ecnu.edu.cn;
ayzhou@dase.ecnu.edu.cn;

Abstract

During long distance transporting for bulk commodities, the trucks need to stop off at multiple places for resting, refueling, repairing or unloading, which are important in transport route planning, called as transport stay hotspots (or *Tshot* for short). Massive waybills and their related trajectories accumulated by the freight platforms enable us to recognize *Tshots* and keep them updated constantly. But due to most of *Tshot*s have varying sizes and are adjacent to each other, it is hard to pinpoint their locations precisely. In addition, to correctly annotate functional tags of *Tshots* that have fewer visiting trajectories is quite difficult. In this paper, we propose a Multi-view Context awareness based transport *Stay hotspot* Recognization fr-amework, called *MCSR*, consisting of *location identification*, *feature extraction* and *functional tag annotation*. To address the mis-detection issue in pinpointing adjacent *Tshots* having various sizes, we design a multi-view clustering based stay area merging strategy by incorporating *distance between road turn-off locations*, *number of visiting trajectories* with *similarity of visiting time distribution*. Further, aiming at the issue of low annotating precision resulted by data scarcity, based upon extracting *behavioral features* and *attribute features* from waybill trajectories, we leverage a *time interval*-aware self-attention network to extract *semantic contextual features* to assist in ensemble learning based annotation modeling correctly. Finally, extensive experiments and case studies are conducted on real steel logistics data to demonstrate the effectiveness and practicability of *MCSR*.

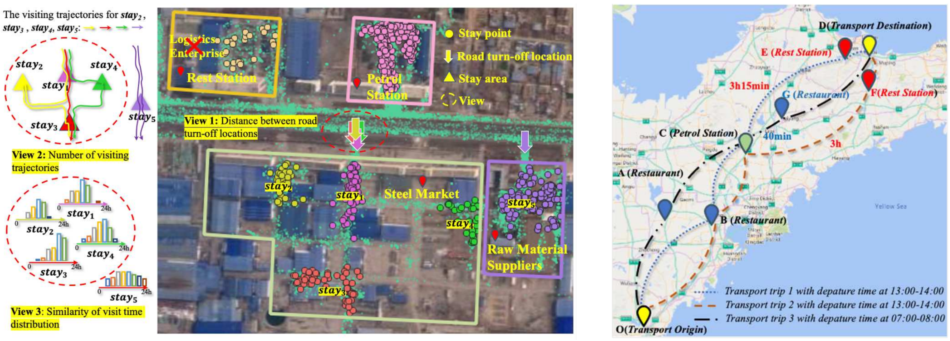
Keywords: transport stay hotspot, multi-view clustering, semantic contextual feature, ensemble learning.

1 Introduction

Bulk commodity transporting is dominated by road transportation and mostly requires for long distance traveling. During transportation, the trucks have to stop off at different places for multiple times for resting, refueling, repairing and unloading, which are called *transport stay hotspots* (or *Tshot*s for short). But due to dynamical changes of *Tshot*s, the truck drivers cannot always find them by the traffic devices or public facilities in *Point Of Interest* (or *POI* for short) data of cities. To be specific, some *Tshot*s may be newly established or relocated. Besides, the truck drivers tend to choose a few abandoned workshops or deserted open spaces as rest stations for economical reasons. Therefore, it necessitates to mine *Tshot*s properly to support transport route planning for road transportation. In past few years, numerous researches have committed to identifying culturally important places or socially meaningful ones according to the frequencies of users visiting such locations and users' travel experiences. But they focus only on the acquisition of the physical location of these significant places for personalized location recommendation [1–3]. In addition, some researches aims at using trajectory to infer places with specific semantics, such as the courier delivery locations [4, 5] and illegal hazardous chemical facilities [6]. Since the truck drivers have various intuitions for choosing stay hotspots during transportation and different stay behaviors at each hotspot, the above works cannot be directly utilized to tackle our proposed *Tshot* recognition issue, which consists of *Tshots*' locations pinpointing and functional tags annotating.

With the widely applications of network freight platforms in bulk logistics field, massive trajectories of the trucks together with waybills are gathered continuously, which offer us an opportunity to mine *Tshot*s. However, we yet need to tackle some unique challenges during the process of location pinpointing and functional tag annotating: **Challenge I. *Tshots* of various scales are adjacent to each other.** *Tshot*s have different scales, i.e. each *Tshot* may contain one or more gathering areas of stay points, called *stay area*s. Besides, some *Tshots* are adjacent to each other, which increases the difficulty for differentiating their respective location and coverage. As shown in Fig.1(a), although *steel market* is adjacent to *raw material supplier*, the former contains four stay point gathering areas (marked as $stay_1$,$stay_2$,$stay_3$ and $stay_4$ respectively), while the latter has only one (i.e. $stay_5$). The case is same to nearby *rest station* and *petrol station*. It easily leads to missed-detection of some small-scale *Tshots* like $stay_4$. **Challenge II. Some *Tshots* have relatively scarce visiting trajectories.** Due to a few *Tshots* are newly established or only known by a narrow group of drivers, the trucks' trajectories occurred on there are scarce. These *Tshots* may be easily misjudged as other classes having more training data by the annotation models that built by the existing methods [7–9]. As illustrated in Fig.1(a), *Rest Station* is misjudged as *Logistics Enterprise* due to it has fewer visiting trajectories of the trucks.

To address the first challenge, in view of that different *Tshots* have their respective locations of turning points on the road (as *road turn-off locations* marked by arrows in Fig.1(a), we first try to merge the *stay areas* that obtained by clustering the stay points based on extracted *road turn-off locations*[10]. But given some *Tshots* (e.g.,


The visiting trajectories for stay2, stay3, stay4, stay5:
stay1
stay2
stay3
stay4
stay5
View 2: Number of visiting trajectories
0
24h
View 3: Similarity of visit time distribution
Logistics Enterprise
Rest Station
Petrol Station
Stay point
Road turn-off location
Stay area
View
View 1: Distance between road turn-off locations
Steel Market
Raw Material Suppliers
D(Transport Destination)
E (Rest Station)
F(Rest Station)
G (Restaurant)
3h15min
C (Petrol Station)
40min
3h
A (Restaurant)
B (Restaurant)
Yellow Sea
O(Transport Origin)
Transport trip 1 with depature time at 13:00-14:00
Transport trip 2 with depature time at 13:00-14:00
Transport trip 3 with depature time at 07:00-08:00

(a) Location Pinpointing

(b) Functional Tag Annotation

Fig. 1: Illustration of Transport Stay Hotspot Recognization

deserted open spaces and open parks) have more than one entrance, the above strategy may result in low precision of location identifying for *Tshots*. It is observed that the trucks will sequentially visit several stay areas of a same *Tshot* in short intervals during a single transportation. It means that any two stay areas having more visiting trajectories of same trucks or similar visiting-time distribution, have a higher probability of belonging to a same *Tshot*. Therefore, based on three views including *Distance between road turn-off locations*, *Number of visiting trajectories* and *Similarity of visiting time distribution*, we design a multi-view clustering based *stay areas* merging strategy to identify the locations of *Tshots*, as illustrated in Fig.1(a).

Aiming at the second challenge, we extract historical visit sequences of *Tshots* between any origin-destination pair and incorporate them into tag annotating process. This is based on the observation that for any origin-destination pair, the trucks which set off at the same time usually visit similar *Tshot* sequence, as transport trip 1 and 2 shown in Fig.1(b). Additionally, we can see from it that for the *Tshot* *E* and *F* having the same functional tag (i.e.*Rest Station*) during the trip 1 and 2, there are similar *Tshot* sequence passing through them, and time intervals of which are the same. Inspired by the above observations, for each *Tshot*, we regard visit sequences of *Tshots* passing through it and related visit time intervals as its *semantic contexts*. On the basis of that, we employ time interval-aware self-attention network to extract *semantic contextual feature* of *Tshot*, and introduce them into the building process of *Tshot*'s tag annotation model.

In general, we propose a Multi-view Context awareness based transport Stay hotspot Recognization framework, called *MCSR*. First, we put forward a *multi-view clustering* based stay area merging strategy to identify *Tshots*' locations, which consists of stay area detection, multi-view fusion based similarity evaluation and stay area merging. Then, based upon extracting *behavioral features*,*attribute features* and *semantic contextual features* of *Tshots*, a multi-classification model based on ensemble learning for *Tshot* annotation is designed. In more detail, the key contributions of our work are summarized as follows:

- We address the issue of *stay hotspot* recognizing for bulk commodity transporting, and then design a multi-view context awareness based framework, consisting of *location identification*, *feature extraction* and *functional tag annotation*.
- Aiming at the issue of missed-detection of neighbouring small-scale *Tshots*, a multi-view clustering based stay area merging strategy is presented to identify *Tshots*' locations, which fuses the views of *Distance between road turn-off locations*, *Number of visiting trajectories* and *Similarity of visiting time distribution*.
- To tackle the low annotating precision issue of *Tshots* brought by scarce visiting trajectories, we not only extract *behavioral features* and *attribute features*, but leverage a time-interval-aware self-attention network to extract *semantic contextual features* and combine them to build an ensemble learning based annotation model.
- We evaluate our proposal based on a large scale of real logistics data set, and observe an average improvement of 14.76% on the *F-measure* metric and 12.89% on the $AIoU$ metric for *location identification*, and 18.39% on the *G-mean* metric and 14.48% on the $mAUC$ metric for *functional tag annotation*, as compared to the state-of-the-art baselines.

The rest of this paper is organized as follows. Section 2 reviews the latest work related to our research. Section 3 provides preliminaries and the problem definition. In Section 4, we outline and analytically study *MCSR* framework. In Section 5, extensive experiments and a case studies are conducted on real datasets to evaluate *MCSR*. Finally, we conclude the paper in the last Section 6.

2 Related Work

In the past decade, the issues of significant place identification [1–7, 11–13] and POI semantic annotation[7–9, 14–19] have attracted wide attentions in academia and industry, and various solutions have emerged accordingly.

Significant Place Identification. Numerous researches attempted to identify the locations of interesting or important places using trajectories according to a certain behavior characteristic of moving objects, e.g., long-time staying or high-frequency turning. Zheng et al. tried to infer interesting locations through density-based clustering[1–3]. Ruan et al. recognized real delivery locations by clustering the stay points nearby the destination location of waybills [4, 5]. Zhu et al. identified candidate locations based on clustering stay points extracted from trucks' trajectories, and then detected illegal chemical facilities by determining whether each of them had loading/unloading events [6, 11]. Besides this, a branch of researches detected road intersections using trajectories according to the characteristics of multiple turning directions and slowing down. Huang et al. detected the road intersections based on clustering convergence points from trajectories [7]. Mao et al. identified road intersections of different scales in terms of heading direction difference and speed variation characteristics of trajectories within various sizes of grid cells [12, 13]. The above approaches usually clustered a number of meaningful trajectory points first and then intuitively regarded the centers of clusters as the locations of significant places. They are unsuitable for identifying large-scale places containing several areas where a certain distance lies between areas, e.g., a logistics park having multiple companies, a large steel mill with many warehouses, etc. To address this issue, we put forward a road turn-off location based destination identification strategy in our previous study [10]. But it still cannot ensure precisely identifying locations of the *Tshots* having multiple entrances such as deserted open spaces.

***POI* Semantic Annotation.** Most of *POI* semantic annotation methods tended to decide functional tags of *POI*s by building a classification model based on extracting discriminative features from check-in data. Ye et al. designed a semantic annotation method by separately building a binary *SVM* model for each tag based upon explicit patterns at individual places and implicit relatedness among similar places [7]. Krumm et al. proposed a *POI* classifier consisting of a forest of boosted decision trees, which is built on such features as the timing of visits and nearby businesses.[8, 9]. He and Hegde respectively exploited the features of users' check-in activities and other behavior data to train a generative probabilistic model to infer tags for *POIs*[14, 15]. To enhance the accuracy of *POI* annotation model, Yang et al. proposed a semi-supervised learning model based on graph embedding to generate discriminative embedding for the places in LBSNs [16]. Zhou et al. presented a tri-adaptive collaborative learning framework to seek for an optimal *POI*-tag score matrix [17]. Manisha et al. proposed a semantic annotation model for location-based social networks through incorporating temporal factors, geographical influence and user-interests [18]. Zhou et al. developed a multi-mode description generator incorporated with a multi-mode encoder and a transformer-based decoder, which generates the descriptions based on *POIs*' reviews and other features [19]. The above mentioned methods mostly attain optimal performance based on extracting behavioral features from sufficient amount of training data. But in bulk logistics field, some *Tshots* have fewer visiting trajectories as compared to the majority of *Tshots*, i.e. the data distribution is uneven. So the aforementioned methods cannot guarantee extracting valuable features from them to build an effective tag annotation model.

3 Problem Definition

In this section, we introduce some preliminary concepts and formalize the issue of *Tshots* recognizing based on trucks' trajectories and waybills.

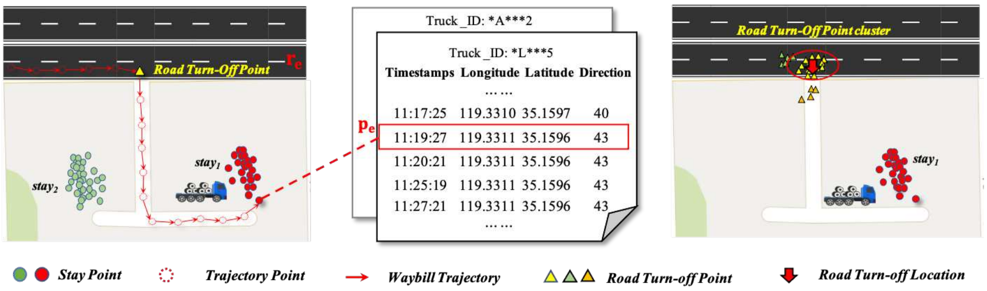
Definition 1 (Trajectory of a Truck)**.** *A trajectory of the truck j refers to a sequence of positional points that chronologically sampled during a time period, denoted as $Tr_j = \{p_1, p_2, \cdots, p_n\}$, where $p_i = (lng, lat, t)$ $(1 \leq i \leq n)$ represents a positional point having the properties of longitude, latitude, and timestamp.*

Definition 2 (Waybill)**.** *A waybill refers to the l-th transport task assigned to the truck j, denoted as a three-tuple $W_j^l = (t_s, t_d, C_{type})$, where t_s is the timestamp when the driver accepts the waybill, t_d is the timestamp of completing unloading confirmed by a truck driver, and C_{type} denotes the type of cargo to be transported.*

A truck trajectory is split into several *waybill trajectories* belonging to different transportation task (or transport trip) based on t_s and t_d of each waybill. The sequence of zero-speed positional points can be extracted from *waybill trajectories*, called as *stay point*.

Definition 3 (Stay Point). *Given a duration threshold* thr_{dur} *and a contiguous zero-speed positional point sequence* $\{p_e, p_{e+1}, \cdots, p_f\}(e < f)$ *extracted from a* waybill trajectory, *the first point* p_e *is viewed as a stay point if the time gap between* p_e *and* p_f *is beyond* thr_{dur}, *and the timestamp of* p_e *is viewed as visiting time.*

Definition 4 (Road Turn-off Point). *Given a stay point* p_e *and waybill trajectory that it belongs to, map matching is performed on* waybill trajectory *to recognize* p_e*'s corresponding turn-off road (denoted as* r_e*), and the last positional point on* r_e *is treated as* p_e*'s related road turn-off point, as illustrated in Fig.2.*



Truck _ID: *A***2

Truck _ID: *L***5

Timestamps	Longitude	Latitude	Direction
……			
11:17:25	119.3310	35.1597	40
11:19:27	119.3311	35.1596	43
11:20:21	119.3311	35.1596	43
11:25:19	119.3311	35.1596	43
11:27:21	119.3311	35.1596	43
……			

Fig. 2: Illustration of Road Turn-off Point

Definition 5 (Road Turn-off Location). *Given a stay area obtained by clustering stay points and its corresponding road turn-off point set, clustering is performed on all road turn-off points, and the center of the largest road turn-off point cluster is treated as that stay area's corresponding road turn-off location, as illustrated in Fig.2.*

As mentioned earlier, we cluster the stay points to generate stay areas, and then merge the stay areas into stay hotspots in terms of their corresponding *road turn-off locations*. Subsequently, for each identified *Tshot*, we obtain the sequence of *Tshots* passing through it and related visit time intervals as its *semantic context*. On the basis of that, we extract the *semantic contextual features* of each *Tshot* for annotation.

Definition 6 (Transport Stay Hotspot). *A transport stay hotspot is a uniquely identified place where the trucks usually stay for resting, refueling, repairing or unloading, which contains the attributes of geographical location* h_{loc} *(represented by longitude and latitude coordinates) and functional tag* h_{tag} *(e.g., rest station, logistics enterprise, etc).*

Problem Definition. Given a collection of trucks' trajectories Tr_s and waybills W_s, our task is to identify each *Tshot*'s location and annotate its functional tag.

4 Overview

As shown in Fig.3, we present a Multi-view Context awareness based transport *Stay hotspot* Recognization framework, called *MCSR*, which consists of (1) *location identification* that identifies the locations of *Tshots* using trajectories and waybills, (2)*feature extraction* that extracts *behavior features*, *attribute features* and *semantic contextual features* for *Tshots*, and (3)*functional tag annotation* that annotates the functional tag of *Tshots* by building an ensemble learning-based model.

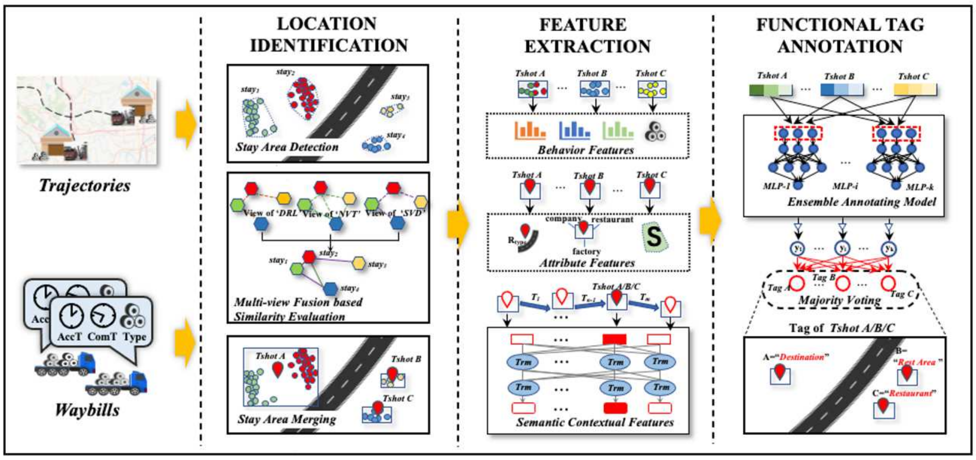


Fig. 3: Overview of *MCSR*

4.1 Location Identification

The main task of *Location Identification* involves *stay area detection*, *multi-view fusion based similarity evaluation* and *stay area merging*, the details of which is given in Algorithm 1.

Stay Area Detection based on *DBSCAN* Clustering. Initially, we split the trajectories of the trucks into *waybill trajectories* according to the start and arrival timestamps of different waybills. Then we extract the stay points from *waybill trajectories* to detect stay areas. To be specific, given that trucks may stay on road due to waiting for traffic lights or traffic jams, we only focus on the trajectory point sequence that keeps zero velocity for a longer period of time, i.e. stay duration is beyond the preset threshold thr_{dur} (here thr_{dur} is empirically set as 8 minutes). We regard the first point in such a sequence as a stay point, and then cluster these stay points using *DBSCAN* method to obtain *stay areas* (as $stay_1$, $stay_2$ shown in Fig.3) (at lines 3-5 in Algorithm 1). To avoid grouping stay points on both sides of the road into the same cluster, we set the cluster radius *eps* as 5 meters in terms of minimum road width that trucks can actually pass. Besides, we set min_{sample} as 5 to ignore the stay areas with a stay frequency less than 5.

Multi-view Fusion Based on Similarity Network Fusion. We extract *road turn-off points* from *waybill trajectory* which its stay point belongs to (according to *Definition 4*), and then cluster *road turn-off points* for each stay area using *Meanshift* method to obtain *road turn-off location* of each stay area. After that, we generate a similarity graph by treating stay areas as the nodes and the distance between their corresponding *road turn-off locations* as the edge weights. Here, the adjacency matrix of the similarity graph is expressed as $W^{TurL} \in R^{m \times m}$ (see formula 1), where m is the number of stay areas, $w_{i,j}^{TurL} \in W^{TurL}$ is the distance between the corresponding *road turn-off locations* of the stay areas $stay_i$ and $stay_j$ $(1 \leq i, j \leq m)$, and ς represents any one view for location identification. Further, we generate similarity graphs for the views about *the number of visiting trajectories* and *the similarity of the visit* *time distribution*, denoted as W^{NumT} and W^{SimD} respectively. Specifically, we count the number of visiting trajectories between stay areas and view their reciprocals as the edge weights of W^{NumT}. In addition, we obtain the visit time distribution of each stay area by counting the visit frequency every two hours within 24-hours of a day, and design a piece-wise function (see formula 2) to calculate the edge weight of W^{SimD} (at line 7 in Algorithm 1).

$$W^{\varsigma\in\{TurL,NumT,SimD\}} = \begin{bmatrix} w^{\varsigma}_{1,1} & w^{\varsigma}_{1,2} & \cdots & w^{\varsigma}_{1,m} \\ w^{\varsigma}_{2,1} & w^{\varsigma}_{2,2} & \cdots & w^{\varsigma}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w^{\varsigma}_{m,1} & w^{\varsigma}_{n,2} & \cdots & w^{\varsigma}_{m,m} \end{bmatrix} \tag{1}$$

$$w^{SimD}_{i,j} = \begin{cases} K\text{-}S(F_i(x), F_j(x)), & if\ dis(stay_i, stay_j) < \gamma \\ 0, & otherwise. \end{cases} \tag{2}$$

where *K-S()* denotes *Kolmogorov-Smirnov test* function. It is used to output the maximum deviation value D between two visiting time distributions by evaluating the similarity between two distributions, here the value range of D is [0,1]. The smaller D's value, the more similar the visiting time distributions are. To avoid similarity evaluation of stay areas that are far apart from, we preset a distance threshold γ and empirically set it as 3 kilometers. Additionally, to normalize the similarity of each view, we use a scaled exponential similarity kernel on similarity graphs, and update their edge weights by $exp\left(-\frac{w^{\varsigma}_{i,j}}{\mu\varepsilon_{i,j}}\right)$, and $\varepsilon_{i,j} = \frac{mean(N_i)+mean(N_j)+w^{\varsigma}_{i,j}}{3}$, where $mean(N_i)$ is the average value of the edge weights between $stay_i$ and each of its neighbors, and μ is a hyperparameter that is set as 0.5.

To fuse three normalized similarity matrices, we employ *similarity network fusion* [20] (or *SNF* for short) to compute the fused matrix (at line 8 in Algorithm 1). To be specific, we define a state and sparse kernel matrix on the stay areas of each view, represented as P^{ς} and S^{ς} respectively, and obtain their values by the following formula:

$$P^{\varsigma}(i,j) = \begin{cases} \frac{w^{\varsigma}_{i,j}}{2\sum_{k\neq i} w^{\varsigma}_{i,k}}, & j \neq i \\ 1/2, & j = i \end{cases}, \quad S^{\varsigma}(i,j) = \begin{cases} \frac{w^{\varsigma}_{i,j}}{\sum_{k\in N'_i} w^{\varsigma}_{i,k}}, & j \in N'_i \\ 0, & otherwise \end{cases} \tag{3}$$

where N'_i is the N nearest neighbors of $stay_i$ in W^{ς}. Then we use S^{ς} as the kernel matrix and start to perform SNF from initial state P^{ς}, and iteratively update the similarity network corresponding to each view according to the following equation.

$$P^{\varsigma}_{t+1} = S^{\varsigma} \times \frac{\left(\sum_{v\neq\varsigma} P^{v}_{t}\right)}{2} \times (S^{\varsigma})^{T} \tag{4}$$

After t steps, we get the fused similarity matrix as $W = \frac{\sum_{\varsigma} P^{\varsigma}_{t}}{3}$.

Stay Area Merging based on *Spectral Clustering*. Since a *Tshot* may have more than one stay areas, we leverage *spectral clustering* method to merge stay areas to obtain *Tshots*. The reason for using spectral clustering is that it is widely applied

Algorithm 1 Transport Stay Hotspot Pinpointing

```
Input: The set of waybills $W_s$ and trajectories $Tr_s$;
Output: The set of $Tshot$ locations $H_{loc}$ and their profiles $H_{prof}$;
1: $H_{loc}, H_{prof} \Leftarrow \emptyset$
2: //Stay Area Detection based on $DBSCAN$ Clutering
3: $WT_s \Leftarrow waybillTrajSplit(Tr_s, W_s)$;
4: $ST_s \Leftarrow stayPointExt(WT_s)$;
5: $\{stay_1, stay_2, \cdots, stay_m\} \Leftarrow DBSCAN(ST_s)$;
6: //Multi-view Fusion based on Similarity Network Fusion
7: $W^{TurL}, W^{NumT}, W^{SimD} \Leftarrow simMatrixCons(\{stay_1, stay_2, \cdots, stay_m\})$;
8: $W \leftarrow SNF(W^{TurL}, W^{NumT}, W^{SimD})$;
9: //Stay Area Merging based on Spectral Clustering
10: $\{h_1, h_2, \cdots, h_k\} \Leftarrow spectralClus(W)$;
11: for $h_i \in \{h_1, h_2, \cdots, h_k\}$ do
12:     $H_{loc} = H_{loc} \cup \{h_i.centroid\}$;
13:     $H_{prof} = H_{prof} \cup \{h_i.profiles\}$;
14: end for
15: return $H_{loc}, H_{prof}$
```

to solve multi-view clustering problem owing to its capability of capturing global structure of the graph [21, 22].

We employ *spectral clustering* method on the similarity graph W to obtain the vector representations of the stay areas in the transformed low-dimensional space by decomposing the eigenvalues of the Laplacian matrix. Then we perform clustering on the vector representations to produce a given number of clusters. More specifically, we leverage *Eigengap heuristic* technique [23] to determine the number of *Tshots* k first, and then generate the representation matrix of the stay areas by the following equation.

$$Y = arg \min_{Y' \in R^{m \times k}} Trace(Y'^{T} L^{+} Y') \qquad s.t. \quad Y'^{T} Y' = I \tag{5}$$

where L^{+} is the normalized Laplacian matrix, denoted as $L^{+} = I - D^{-1/2} W D^{-1/2}$, here D is a diagonal matrix of W, $Trace()$ denotes the trace of a matrix, and Y is obtained scaled partition matrix. We take each row $y_i \in Y$ as the vector representation of $stay_i$ in the transformed low dimensional space, and perform hierarchical clustering on such vector representations to obtain k stay area clusters (at line 10 in Algorithm 1).

Next, we merge the stay areas that belong to a same cluster to obtain a *Tshot* and regard the centroid of such a cluster as that *Tshot*'s location. Besides, we extract the features from historical data and then generate the profile for each *Tshot*, which includes *Visiting time distribution*, *Average and median stay duration*, *Number of cargo types*, *Distribution of Stop frequencies for single transportation* and *Acreage of stay hotspot* (at lines 12-13 in Algorithm 1).

4.2 Feature Extraction

We extract three types of features from *Tshots*' historical visiting trajectories, including *behavioural features*, *attribute features*, and *semantic contextual features*.

Behavior Features. They can reveal behavioural patterns of the trucks at each *Tshot*, which involve the following features: (1)*visiting time distribution* (denoted as F_{vt}) that is used to calculate the time distribution of the trucks visiting each *Tshot* by discretizing 24 hours into hourly time-slots. As shown in Fig.4, different types of *Tshots* exhibit distinct visiting time distributions, e.g., the visiting time of the logistics company is usually concentrated during the daytime, while that of the rest station is not the same; (2)*Average and median stay duration* (denoted as F_{sd}), which is derived by calculating the mean and median of the stay duration of each *Tshot* respectively and then concatenating them together. The reason for choosing it as a feature is that the trucks have different stay durations for *Tshots* of various types due to distinct demands of the drivers. As shown in Fig.5, the stay duration of the trucks staying at the petrol station is about 15 to 30 minutes, while that of rest areas is about dozens of minutes or even hours; (3)*Number of cargo types* (denoted as F_{ct}), which is used to calculate the number of cargo types related to a *Tshot* in historical waybills. For example, the types of cargoes related to a logistics company only involves those of its business range, while those related to the gas station or rest station may include other types of cargoes; (4) *Distribution of Stop frequencies for single transportation* (denoted as F_{sf}), which is obtained by counting the distribution of the number of *waybill trajectories* corresponding to the frequency of stops in *Tshot*, expressed as a vector $[fre_1, fre_2, \cdots, fre_l]$, where $fre_i (1 \leq i \leq l)$ denotes the number of *waybill trajectories* that have i stay points at the *Tshot*, and l is maximum historical frequency of the trucks stop at that *Tshot* in a single transportation. The reason for choosing it as a feature due to that the trucks usually make multiple stops at a petrol stations or logistics companies during single transportation, but only once when visiting the places like rest stations.

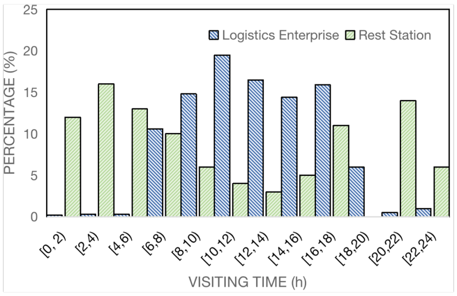


Fig. 4: Visiting Time Distribution

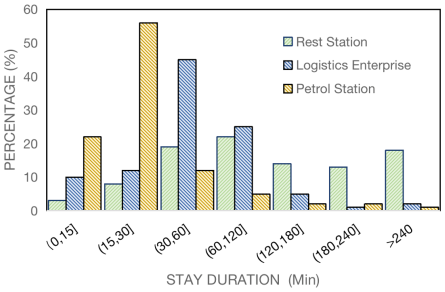
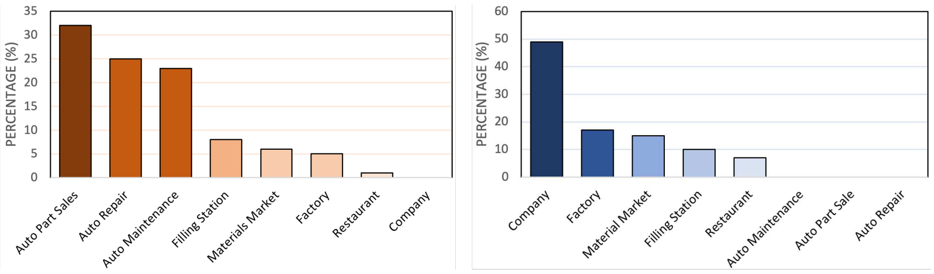


Fig. 5: Stay Time Distribution

Attribute Features. They reveal *Tshot*'s geographical properties, including (1) *Acreage of stay hotspot* (denoted as F_{as}), which is derived by extracting the polygon covering all the stay points in a *Tshot* by using convex hull algorithm [24] first, and then calculating the acreage of that polygon. The reason for considering it as a feature is that the acreages between different type of *Tshots* exhibit significant differences, e.g., the rest stations and logistics enterprises always have larger acreages than others; (2) *Type of neighboring road* (denoted as F_{nr}), which is the type of *Tshot*'s nearest road segment obtained by performing map matching. The reason we choose this as one attribute feature is based on actual observations, e.g., the catering service area tends to be located nearby the motorway to satisfy the demand of truck drivers' resting, while the logistics enterprises locate beside low-level roads such as provincial highway or national highway for cargo loading convenience; (3)*Distribution of nearby POI categories* (denoted as F_{pc}), which is expressed by a vector consisting of 21 *POI* category, and obtained by counting the number of *POIs* corresponding to each type within the range of 100 meters, 200 meters, and 500 meters near *Tshot* respectively. As shown in Fig.6, a maintenance station locates in the region that accompanies with a large number of *POIs* related to auto repairing, and a logistics enterprise locates within the area having many factories.



(b) Maintenance Station Instance (a) Logistics Enterprise Instance

Fig. 6: Distribution of Nearby *POI* Categories

Semantic Contextual Features. As mentioned earlier, the *Tshots* with similar semantic context (i.e. the sequence of *Tshots* passing through it and the visit time interval of those *Tshots*) usually have the same functional tags, we further extract semantic context features for each *Tshot* from historical waybill trajectories. In view of the advantage of *BERT* in word context understanding, we incorporate *BERT* with the time interval-aware self-attention network [25] to extract semantic contextual features by treating *Tshots* as the words and the sequence of *Tshots* as the sentences.

To be specific, we first extract the sequence of *Tshots* that passing through each *Tshot* and the visit time intervals of every element in each sequence from *waybill trajectory* data. Then we stack L time interval-aware self-attention layers on *Tshot* sequences to generate context representation of *Tshot*. The implementation details are shown in Fig.7, at the bottom of stacks, the input representation of i_{th} *Tshot* of the *waybill trajectory* is obtained as $h_i^0 = v_i + p_i$, where v_i is the d-dimensional embedding for *Tshot*, p_i is the d-dimensional positional embedding for position index i. Initial representation of *Tshot* sequence is generated by concatenating the embeddings of *Tshots* in it, expressed as $H^0 = h_1^0 \parallel h_2^0 \parallel \cdots \parallel h_{|H|}^0$, here $|H|$ is the length of the *Tshot* sequence. Subsequently, we iteratively compute the hidden representation of each layer for *Tshot* sequence. Specifically, we apply the time interval-aware self-attention network on each layer $l \in [1, L]$ as follows:

$$H^l(Q,K,V) = ||_{h=1}^{|h|}\, softmax(\frac{Q_h \cdot {K_h}^T}{\sqrt{d/|h|}} + \tilde{\triangle})V_h \tag{6}$$

where $|h|$ is the number of heads, $Q_h = H^{l-1}W_h^Q$, $K_h = H^{l-1}W_h^K$, $V_h = H^{l-1}W_h^V$, and H^l denotes the output of layer l. The projections matrices for each head $W_h^Q, W_h^K, W_h^V \in R^{d\times d/|h|}$ are learned parameters. Additionally, $\tilde{\triangle} \in R^{|H|\times|H|}$ represents an adaptive time interval matrix. Each element $\tilde{\triangle}(i,j)$ measures the impact of the time interval between the i^{th} and j^{th} *Tshot*, which is calculated as follows:

$$\tilde{\triangle}(i,j) = (leakyReLU(\delta_{i,j}^{'}\omega_1))\omega_2^T \tag{7}$$

$$\delta_{i,j}^{'} = 1/log(e + \delta_{i,j}) \tag{8}$$

where ω_1 and ω_2 are learnable parameters, $LeakyReLU$ is an activation function whose negative input slope is 0.2, and $\delta_{i,j}$ is a relative time interval for any two *Tshots* in sequence.

After that, we apply a position-wise feed-forward Network (or FFN for short) to H^l. FFN consists of two layers of linear transformations and $ReLU$ activation, which is defined as $H^l = (ReLU(H^l W_F^1 + b_F^1))W_F^2 + b_F^2$. $W_F^1, W_F^2 \in R^{d\times d}$, here $b_F^1, b_F^2 \in R^d$ are learnable parameters. Then we obtain the final output representation $H^L \in R^{|H|\times d}$ for *Tshot* sequence after stacking L layers of time interval-aware self-attention module, and take each row $h_i \in H^L$ as semantic contextual representation of the i_{th} *Tshot* of *waybill trajectory*. Consider that multiple trucks' *waybill trajectories* may pass through a same *Tshot*, several different semantic contextual representations of each *waybill trajectory* can be generated. Therefore, we calculate average vector of these semantic contextual representations to obtain the semantic contextual features (denoted as F_{sc}) of a *Tshot*, as illustrated in Fig.7.

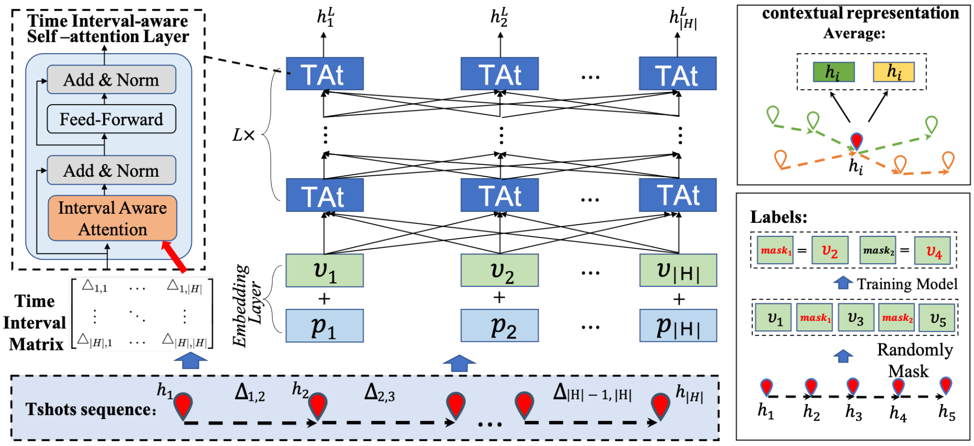


Fig. 7: Illustration of Semantic Contextual Features Extraction

Furthermore, we apply the masked language task for training the time interval-aware self-attention network. As shown in Fig.7, for each training step, we randomly mask ρ proportion of all *Tshots* for a *waybill trajectory*, and then predict original *ids* of the masked *Tshots*. To be specific, we apply a two-layer feed-forward network with $GELU$ activation in between to produce an output distribution over masked *Tshots* as $P(h) = softmax(GELU(h_{mask}W^P + b^P)E^T + b^O)$, where $W^P \in R^{d\times d}$ is a learnable projection matrix, $b^P, b^O \in R^d$ are bias terms, $E \in R^{|Hs|\times d}$ denotes the embedding matrix of *Tshots*, and $|Hs|$ is the number of *Tshots*. Finally, we use the cross-entropy loss between masked *Tshots* and predicted values as optimal objective, as shown below.

$$L = -\frac{1}{|Set_{mask}|} \sum_{h_{mask}\in Set_{mask}} logP(h_{mask} = h|H_{mask}) \tag{9}$$

where $Hmask$ is masked version for the *Tshot* sequence, Set_{mask} is the masked *Tshot* set, h is true *Tshot* of masked one h_{mask}.

4.3 Functional Tag Annotation

Inspired by the idea that ensemble learning [26–32] can improve overall performance by combining the decisions from multiple base models for a task, we leverage ensemble learning technique for annotations of *Tshots*' functional tags. To search for an optimal sample subset group for learning multiple annotation models, we employ genetic algorithm [33] on the *Tshot* samples (i.e. training data). As depicted in Fig.8, we extract k' sample subsets from *Tshot* samples first, and use each subset for training an annotation model. Then we continuously evolve to generate new sample subset group based on fitness evaluations for the annotation models. After a given number of evolutions, we obtain k' annotation models that learnt from the evolved sample subset group as an ensemble (the implementation details are given in Algorithm 2). Finally, we employ that ensemble to jointly infer the functional tag of each *Tshot*.

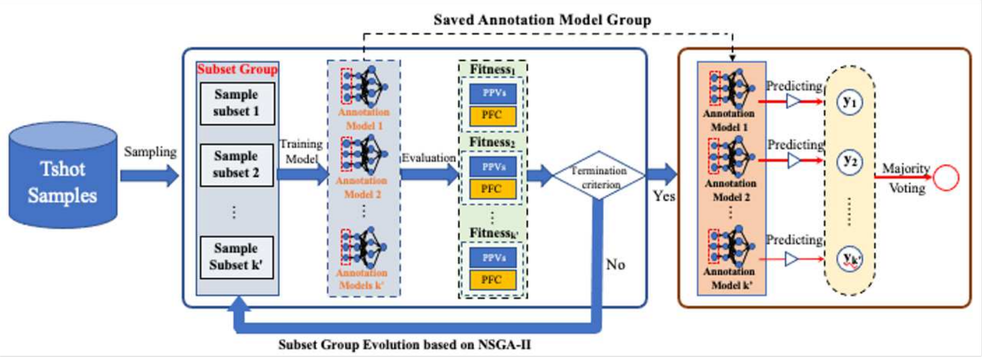


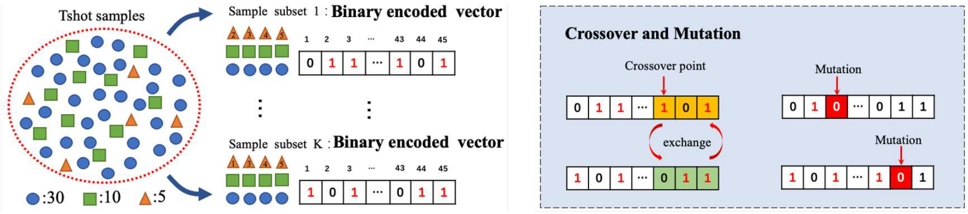
Fig. 8: Ensemble Learning Framework for *Tshot* Annotation

Sample Subsets Group Initialization. We initialize the evolution by randomly sampling $k^{'}$ equal-sized sample subsets (at lines 3-4 in Algorithm 2). The size of each subset is determined by the number of samples of the minority $Tshot$ type. As shown in Fig.9(a), take an sample set having 3 $Tshot$ types and 5 samples at the minority type as an example, the size of the subset should be 12 (80%×5×3). Each subset is treated as an individual for evolution, and then used for training an annotation model. Additionally, we represent each sample subset as a binary-encoded vector, whose length is the number of total $Tshot$ samples, here the presence or absence of a $Tshot$ sample is represented by a 1 or 0 respectively.

Algorithm 2 Ensemble Generation of Transport Stay Hotspot Annotation

```
Input: $Tshot$ samples $Set_h = \{h_1, h_2, \cdots, h_{\#sample}\}$;
Output: Annotation model group $Ens_{MLP}$;
 1: $Ens_{MLP} \Leftarrow \emptyset$;
 2: //Sample Subset Group Initialization
 3: $Subs^0 = \{sub_1^0, sub_2^0, \cdots, sub_{k'}^0\} \Leftarrow sample(Set_h)$;
 4: $MLPs^0 = \{M_1^0, M_2^0, \cdots, M_{k'}^0\} \Leftarrow learnMLP(Sub_s^0)$;
 5: //Sample Subset Group Evolution
 6: for $i \in [0, w]$ do
 7:     $Fitns \Leftarrow fitnessEva(MLPs^i, Set_h)$;
 8:     $Subs_{off} \Leftarrow genOper(Subs^i, Fitns)$;
 9:     $MLPs_{off} \Leftarrow learnMLP(Subs_{off})$;
10:     $Fitns_{off} \Leftarrow fitnessEva(MLPs_{off}, Set_h)$;
11:     $Subs_{new} \Leftarrow NDSort(\{Subs^i \cup Subs_{off}\})$;
12:     $MLPs^{i+1} = \{M_1^{i+1}, M_2^{i+1}, \cdots, M_{k'}^{i+1}\} \Leftarrow learnMLP(Subs_{new})$;
13:     $i = i + 1$;
14: end for
15: $Ens_{MLP} = MLPs^{w+1}$;
16: return $Ens_{MLP}$
```

Fitness Evaluation. In view of that an effective ensemble consisting of a set of models should produce different predictions on parts of the input space while obtaining high-accuracy prediction results [26, 34], we introduce PPV[26] and PFC[35] to evaluate the evolutionary quality of sample subsets. Given a subset $sub_j (j \leq k^{'})$ and annotation model learned from sub_j, PPV is used to evaluate the prediction accuracy of the annotation model for $Tshot$ type i, calculated by $PPV_i^j = \frac{\#true\_pos_i^j}{\#sample_i}$, where $\#true\_pos_i^j$ denotes the number of $Tshots$ with type i correctly inferred by the annotation model among all $Tshot$ samples, and $\#sample_i$ is the number of $Tshots$ of type i in all $Tshot$ samples. It is obvious that each sample subset is associated with multiple $PPVs$ due to various types of $Tshots$. PFC is used to evaluate the diversity of the annotation model, calculated as $PFC^j = \frac{1}{k'-1}\sum_{o\neq j}\frac{\sum_{n=1}^{\#sample} I(gp_n^j, gp_n^o)}{\#false\_pos^j+\#false\_pos^o}$, where $\#sample$ is the number of total $Tshot$ samples, and gp_n^j is the output of annotation model j with the $Tshot$ sample. Indicator function $I()$ returns 1 if the inferred outputs are different, or 0 otherwise. $\#false\_pos^j$ denotes the number of predictions errors of annotation model j for all $Tshot$ samples. It will return values between 0 and 1, and the higher the PFC^j, the better the diversity.



(a) Sample Subsets Initialization

(b) Crossover and Mutation

Fig. 9: Evolution of Sample Subsets

Sample Subset Group Evolution. We employ Non-dominated Sort Genetic Algorithm II[33] to iteratively evolve sample subsets to produce an optimal annotation model group (at lines 6-14 in Algorithm 2). For each evolution, we first randomly select 3 sample subsets from current subset group and extract optimal pair according to $PPVs$ using non-dominated sorting technique [36]. If a tie occurs, the one having highest PFC wins. Then we use genetic operators on this pair to obtain new subsets for the offspring subset group. As shown in Fig.9(b), for each selected pair, we implement the crossover operation by using one-point crossover technique [37] to exchange parts of the binary encoded vectors, and then perform the mutation operators to select a certain percentage of vector values to be varied. We iteratively perform the above steps until $k^{'}$ offspring subsets are obtained (at lines 7-9 in Algorithm 2). Finally, we employ non-dominated sorting technique to extract optimal $k^{'}$ sample subsets from current subset group and offspring subset group as the subset group generated by an evolution (at lines 10-12 in Algorithm 2). After a given number of evolutions, $k^{'}$ annotation models learned from the evolved subset group are obtained (at lines 15-16 in Algorithm 2).

Owing to excellent performance of MLP on multi-classification issue[16, 17], we leverage MLP as the annotation model and obtain $k^{'}$ $MLPs$ as the ensemble for annotation. Specifically, for each $Tshot$, $k^{'}$ $MLPs$ accept their respective features as inputs, and then output $k^{'}$ functional tags. In the end, the majority of the results is viewed as $Tshot$'s functional tag.

5 Experimental Evaluation

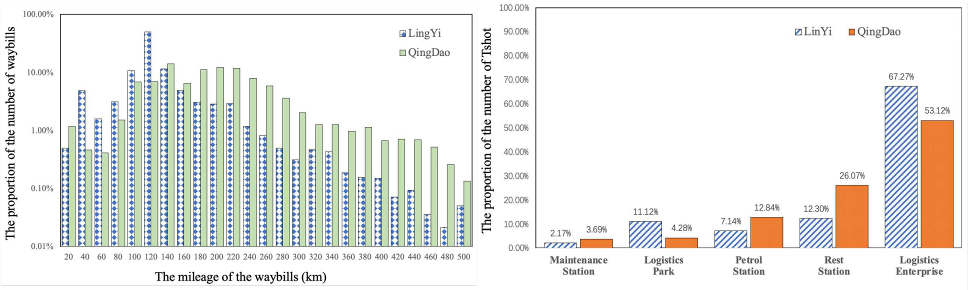
In this section, various experiments will be conducted based on real-word logistics datasets to evaluate the superiority of *MCSR* in *location identification* and *functional tag annotation*.

5.1 Datasets and Settings

Datasets. We utilize a real dataset of 4 months (from Nov. 1^{st}, 2020 to Mar. 1^{st}, 2021) from Steel Logistics Technology Co., Ltd. The Dataset includes the trajectories and waybills of two transportation routes departing from *Rizhao* City in Shandong, China. They are divided into two datasets according to the cities of transport destinations (hereafter termed *QingDao*/*LinYi*), whose details are shown in Table 1. The trajectory point record consists of *timestamps* and *latitude* and *longitude* coordinates. The waybill record consists of *driver ID*, *cargo type*, *start and completion timestamps*. Futhermore, the road network with 652,603 vertices and 1,630,544 edges is obtained from *OpenStreetMap* for trajectory map matching. It is worth noting that the geographical areas and functional tags of 2,573 *Tshots* are carefully manually labeled. For the former, we generate the minimum bounding rectangular of each *Tshot* as its geographical area. For the latter, we assign a functional tag such as *logistics enterprise*, *rest station*, *petrol station*, *logistics park*, and *maintenance station* to each *Tshot*. The mileage distribution of the waybills and the volume distribution of *Tshots* are shown in Fig.10. As observed, the *Tshot* volumes of different functional tags in both datasets exhibit imbalanced distribution, and this is more obvious for *QingDao* probably due to its longer transportation mileage.

Table 1: Statistics of dataset

Dataset	# of trajectory points	# of *waybills*	# of *Tshots*
QingDao	243,528,247	184,144	1,512
LinYi	232,525,232	262,621	1,061



(a) Mileage Distribution of waybills (b) Distribution of Tshot

Fig. 10: Dataset Descriptions

Baseline Methods. To evaluate the benefits of our proposal, we single out several contrast approaches, including some significant place identification methods and *POI* semantic annotation methods.

- *MRInf* [38] applies *DBSCAN* clustering algorithm and proposes a parameter selection method to cluster *constrained convergence points* to infer location of major road intersections. We adapt this method and its parameter selection strategy to cluster stay points to detect *Tshots*.

- *DTInf* [5] applies *hierarchical* clustering algorithm, clustering the courier annotation locations to recognize the actual delivery location. Here we use this method to cluster stay points to detect *Tshots*.
- *TDCM*[10] proposes a stay area merging strategy to infer the location of the transport destination. It first clusters stay points to detect stay areas, then infers the *road turn-off locations* for each stay area by clustering *road turn-off points*. Finally, It merges stay areas based on the distance between *road turn-off locations* to generate *Tshots*.
- *PPE* [16] is a *POI* semantic annotation method which utilize both unlabeled and labeled data to jointly enhance the representation of each place through graph embedding.
- *TACL* [17] is a *POI* tag refinement method based on a tri-adaptive collaborative learning framework, which aims to automatically fill in the missing tags as well as correct noisy tags for *POI*.
- *HAP-SAP* [18] uses multi-variate Hawkes process to model the human mobility patterns, this work associate a category to each check-in and employ expectation maximization procedure to infer the missing categories.
- *SAP* [7] is a semantic annotation algorithm based *SVM*, which aims to automatically annotate all places with semantic tags in location-based social networks.

Among them, we evaluate the former three methods for comparison purpose of verifying the effectiveness of our *location identification* method, and choose the latter four methods as contrast methods to demonstrate the effectiveness of our *functional tag annotation* method.

Evaluation Metrics. For the *location identification* of our *MCSR*, we utilize the *Intersection over Union* (or IoU for short) to measure the degree of overlap between the identified *Tshot* area and the labeled geographical area. The IoU is calculated as $\frac{ACR_{inter}}{ACR_{det}+ACR_{tru}}$, where ACR_{inter} denotes acreage of overlapping area between the identified area and labeled geographical area, ACR_{det}, ACR_{tru} denote acreage of the identified area and labeled geographical area. IoU includes two forms: *average* IoU *(A*IoU*)* and *global* IoU *(G*IoU*)*. The former calculates IoU for each *Tshot* separately, and then averages the IoU of all *Tshots*. The latter treats all stay hotspots as a whole. In addition, we regard labeled geographical area with $IoU \geq 0.5$ as successfully identified, and then calculate the *Precision*, *Recall* and *F-measure* of *Tshot* identification. For the *functional tag annotation* of our *MCSR*, we utilize $F1$-*macro*, $F1$-*micro*, *G-mean* and $mAUC$ as evaluation metrics. Formally, $F1$-*macro* is calculated as $\frac{1}{n_{type}} \sum_{i=1}^{n_{type}} F\text{-}measure_i$, where n_{type} is the number of *Tshot* types, $F\text{-}measure_i$ denotes the *F-measure* of the *Tshot* annotation for type i. $F1$-*micro* is calculated as $\frac{\sum_{i=1}^{n_{type}} \#true\_pos_i}{\sum_{i=1}^{n_{type}} \#sample_i}$, where $\#true\_pos_i$ denotes the number of *Tshots* with type i correctly annotated, $\#sample_i$ denotes the amount of *Tshots* of type i. *G-mean* is calculated $\left(\prod_{i=1}^{n_{type}} \frac{\#true\_pos_i}{\#sample_i}\right)^{\frac{1}{n_{type}}}$. Additionally, $mAUC$ is an extension of AUC for multi-classification problems[39], which is also used to evaluate our annotation task.

Experimental Settings. We split each dataset into a training set, a validation set and a test set with a splitting ratio of 7:2:1. All experiments are conducted on GPU-CPU platform with Tesla V100. The program and baselines are implemented in Python 3.8. The main hyper-parameters settings of our proposal are described as follows: For semantic contextual features extraction, we train the time interval-aware self-attention model using Adam algorithm under the rate of 0.01, and set the epoch as 100. Additionally, we set *Tshot* embedding size d to 256, the stack layers L to 6, the attention heads $|h|$ as 8, and the mask ratio ρ as 15%. For functional tag annotation, we set the ensemble size $k^{'}$ as 7, the number of iterations w as 500.

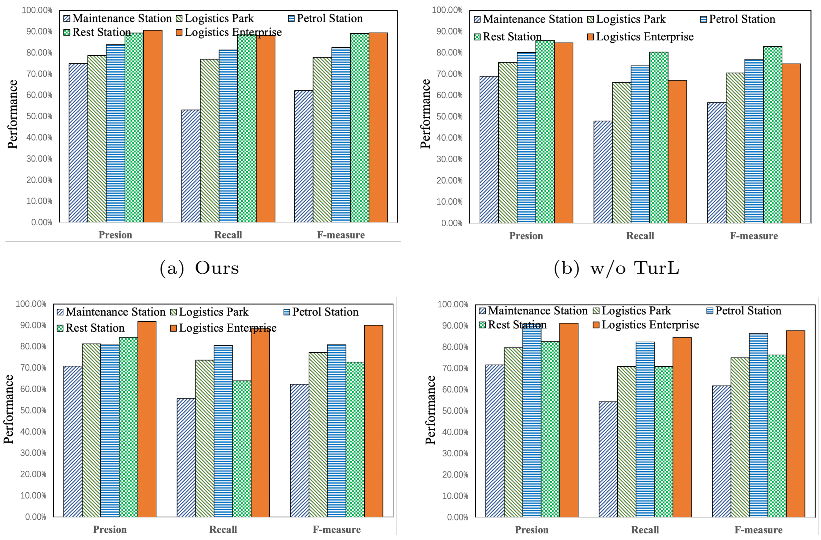
5.2 Location Identification

Overall Peformance. Table 2 shows the overall performance of *location identification* of *MCSR*. From the performance comparison, we find our proposal outperforms all baselines. First of all, *MRInf*, *DTInf* perform poorly on all metrics. They cluster stay points and directly treat each cluster as a *Tshot*, which easily leads to the misidentification of adjacent *Tshots*. Secondly, *TDCM* is suboptimal because it only considers the distance between *road turn-off locations* of stay areas, and has poor performance for *Tshots* with more than one entrance. Finally, *QingDao* has better performance than *LinYi*. We think the possible reason is that *QingDao* has more logistics enterprises that are close to each other.

Table 2: Overall Effectiveness Evaluation of Location Identification

Dataset		*QingDao*					*LinYi*				
Metric(%)		*Precision*	*Recall*	*F-measure*	*AIoU*	*GIoU*	*Precision*	*Recall*	*F-measure*	*AIoU*	*GIoU*
Baseline	MRInf	68.74	53.24	60.01	45.83	56.87	63.82	51.55	57.03	43.71	49.12
	DTInf	67.29	56.87	61.64	47.73	55.93	65.35	49.95	56.62	42.37	51.68
	TDCM	79.23	75.46	77.30	61.07	63.25	74.56	60.22	66.63	54.98	56.04
Variant	*w/o TurL*	82.69	74.27	78.25	66.85	71.07	84.67	63.25	72.59	63.38	67.68
	w/o NumT	88.05	81.87	84.85	73.02	75.38	88.10	78.51	83.03	67.31	72.25
	w/o SimD	89.09	83.20	86.04	71.65	74.25	87.04	74.74	80.42	67.47	72.52
Ours		**89.11**	**88.75**	**88.93**	**73.25**	**75.73**	**87.65**	**81.62**	**84.52**	**68.57**	**73.97**

Ablation Study. In table 2, we conduct the ablation study by replacing our *MCSR* with three vatiations, namely *w/o TurL*, *w/o NumT* and *w/o SimD* to evaluate the effectiveness of different views for *Tshot* identification. In *w/o TurL*, we remove the view of *Distance between road turn-off locations*. Similarly, in *w/o NumT* and *w/o SimD*, we remove the views of *Number of visiting trajectories* and *Similarity of visiting time distribution* respectively. The results show that each view is effective for the identification of *Tshots* as the performance decrease for all variants, and the view of *Distance between road turn-off locations* is the most critical one. Additionlly, we visualize the performance of each type of *Tshot* for the variants in Fig.11. The large drop in performance of the *logistics enterprise* type in Fig.11(b) indicates that the view of *Distance between road turn-off locations* has a more important impact on the identification of *logistics enterprises*. Fig.11(c) and 11(d) show that the other views have a more important impact on the identification of *rest stations*.



(a) Ours

(b) w/o TurL

(c) w/o NumT

(d) w/o SimD

Fig. 11: Ablation Study for Location Identification

5.3 Functional Tag Annotation

Overall Performance. Table 3 shows the overall performance of *functional tag annotation* of *MCSR*. From the performance comparison, we find our proposal outperforms all baselines. *PPE* and *HAP-SAP* perform the worst because they employ semi-supervised learning strategies to annotate *Tshots* with missing tags, which are not suitable for annotate scenarios for where a large number of *Tshot* functional tags are missing. In addition, *SAP* and *TACL* perform suboptimally, and we think the reasons are twofold: 1) They only extract statistical features based on historical *waybill trajectories*, and build classification models to infer the functional tags of each *Tshot*. This performs poorly for annotating *Tshots* with scarce visiting trajectories. 2) Our proposal applies an ensemble learning strategy for annotations of *Tshots*' functional tags. It can improve the annotation accuracy for the *Tshot* types with relatively small data volume. Finally, *QingDao* has higher annotation accuracy than *LinYi*. We think the reason is that the latter has a more serious imbalanced distribution of various types of *Tshots* than the former.

Table 3: Overall Effectiveness Evaluation of Functional Tag Annotation

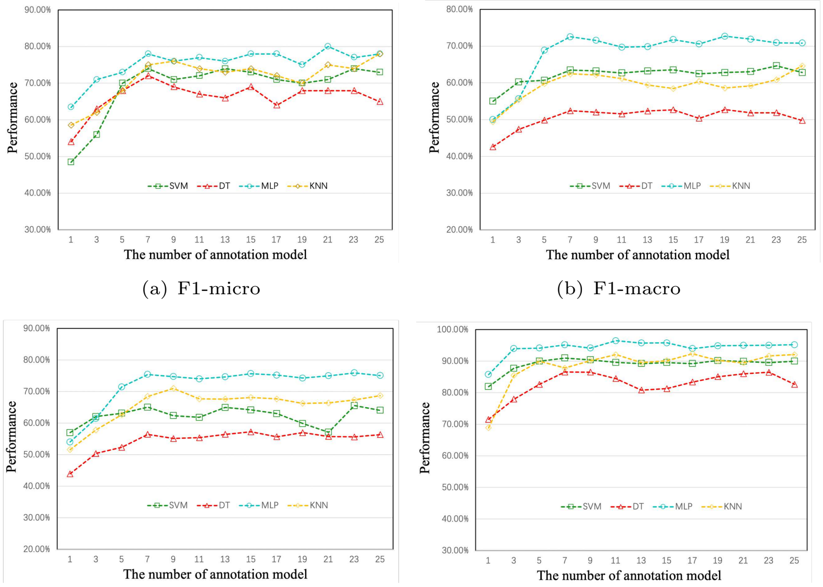
Dataset	*QingDao*				*LinYi*			
Metrics(%)	*F1-micro*	*F1-macro*	*G-mean*	*mAUC*	*F1-micro*	*F1-macro*	*G-mean*	*mAUC*
PPE	58.62	49.45	51.61	68.89	54.38	42.62	43.92	69.57
TACL	68.17	49.88	52.33	82.66	63.13	47.36	50.36	77.93
SAP	69.72	52.68	57.24	79.31	66.24	52.36	56.41	78.81
HAP-SAP	63.45	50.04	54.01	85.74	48.46	55.01	56.96	81.94
Ours	**78.77**	**72.23**	**75.07**	**95.18**	**77.58**	**73.84**	**75.91**	**95.02**

Ablation Analysis of Features. We evaluate the contribution of individual feature to the performance of *Tshot* annotation. The results are shown in Table 4: for each feature, we train a new annotation model on the same dataset, using all features except for that. We then report the relative change of the performance metrics of each new model with respect to the original model that uses all features. These results demonstrate all of the features that we use capture nonredundant information about annotation model, and the *semantic contextual features*(F_{vt}) are the most important ones.

Table 4: Relative Changes of Performance Metrics For Each Feature

Dataset		*QingDao*				*LinYi*			
Metric(%)		*F1-micro*	*F1-macro*	*G-mean*	*mAUC*	*F1-micro*	*F1-macro*	*G-mean*	*mAUC*
Behavior	F_{vt}	+4.92	+7.36	+6.96	+5.05	-3.26	+1.17	+1.26	+2.88
	F_{sd}	+3.14	+3.38	+6.62	+3.38	+2.26	+1.79	+1.38	+2.16
	F_{ct}	+1.02	+1.18	+6.34	+3.13	-2.73	-0.44	+0.26	+0.73
	F_{sf}	+4.08	+4.64	+7.43	+3.14	+4.22	+0.55	+4.98	+2.41
Attri	F_{as}	+7.01	+10.18	+13.61	+1.22	+1.48	-2.67	+12.54	+2.46
	F_{nr}	+6.31	+5.43	+7.43	+2.82	-1.29	-2.42	+10.58	+2.39
	F_{pc}	+7.39	+2.58	+12.71	+4.76	+3.18	+8.36	+12.53	+5.84
F_{sc}		**+14.62**	**+15.78**	**+21.07**	**+9.44**	**+10.41**	**+4.98**	**+9.81**	**+4.88**

Hyperparameter Selection of Annotation Model. Given that the ensemble size $k^{'}$ affects the performance of the annotation, we determine the ensemble size by varying $k^{'}$ from 1 to 25 in steps of 2. In addition, in order to verify the effectiveness of *MLP* for *Tshot* annotation, we replace *MLP* with *support vector machine* (*SVM*), *K-nearest-neighbor* (*KNN*) or *decision tree* (*DT*). The results of $F1$-*micro*, $F1$-*macro*, *G-mean* and *mAUC* are reported in Fig.12(a)-12(d) respectively. They all show that these



(a) F1-micro

(b) F1-macro

(c) G-mean

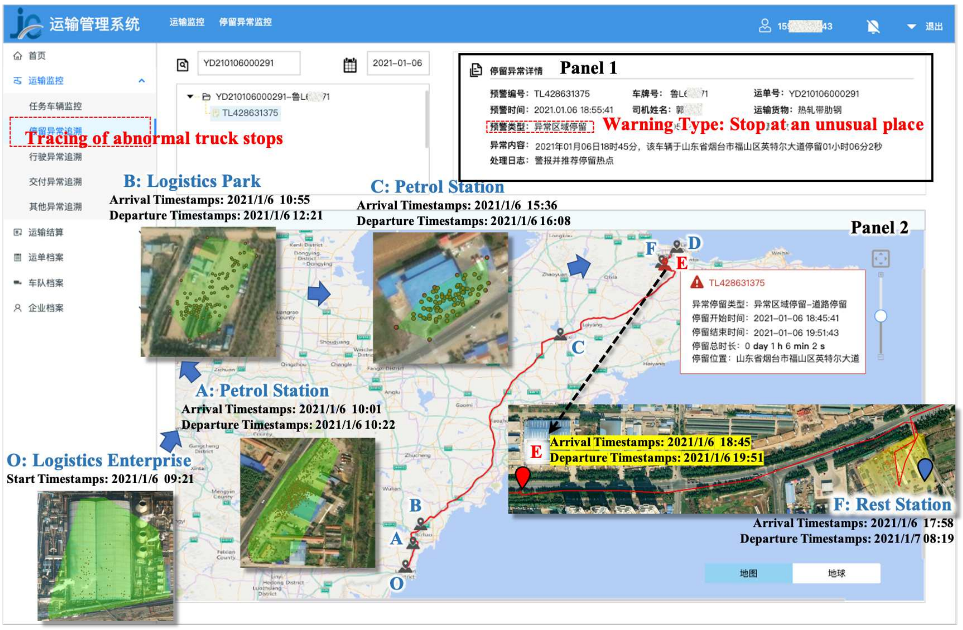
(d) mAUC

Fig. 12: The effect of hyperparameters on annotation model

metrics first rises with the $k^{'}$ up to 7 and then converges. The reason is that the gains can be achieved by incorporating additional classifiers become progressively smaller as the ensemble grows. In addition, it is not difficult to find that *MLP* performs better than other models on all metrics, the reason being that *MLP* is more suitable for solving multi-classification problems with limited training set. In summary, we choose *MLP* as the annotation model for the ensemble and set $k^{'}$ to 7 as it is a convergence point.

5.4 Case Study

Our framework has been applied to a bulk logistics platform to serve applications such as transportation monitoring and route planning. Fig.13 shows an interface for managers to trace the historical abnormal stop behaviors of the transportation trip. The user clicks the "*Tracing of Abnormal Truck Stops*" button (highlighted in red box) to query the historical abnormal stop events. As can be seen from *Panel 1*, an abnormal stop event (marked *E*) that occurred at 18:45 on January 6, 2021 was promptly warned by the system. All stop events (marked *O, A, B, C, E, F, D*) of the transport trip to which E belongs are visualized in *Panel 2*. As can be seen from the Fig.13, the truck was warned by the system because it stopped on the road (i.e. *E*). Then, the system recommends a nearby *rest station* (i.e. *F*) to the truck for it to stop and rest. The subsequent *waybill trajectory* shows that the driver accepted the recommended result. In addition, other stop events occurred in recognized *Tshots* and were not alerted by the system.



预警编号：TL428631375　车牌号：鲁L[redacted]71　运单号：YD210106000291
预警时间：2021.01.06 18:55:41　司机姓名：郭[redacted]　运输货物：热轧带肋钢
预警类型：异常区域停留
异常内容：2021年01月06日18时45分，该车辆于山东省烟台市福山区英特尔大道停留01小时06分2秒
处理日志：警报并推荐停留热点

TL428631375
异常停留类型：异常区域停留-道路停留
停留开始时间：2021-01-06 18:45:41
停留结束时间：2021-01-06 19:51:43
停留总时长：0 day 1 h 6 min 2 s
停留位置：山东省烟台市福山区英特尔大道

O: Logistics Enterprise
Start Timestamps: 2021/1/6 09:21

A: Petrol Station
Arrival Timestamps: 2021/1/6 10:01
Departure Timestamps: 2021/1/6 10:22

B: Logistics Park
Arrival Timestamps: 2021/1/6 10:55
Departure Timestamps: 2021/1/6 12:21

C: Petrol Station
Arrival Timestamps: 2021/1/6 15:36
Departure Timestamps: 2021/1/6 16:08

E
Arrival Timestamps: 2021/1/6 18:45
Departure Timestamps: 2021/1/6 19:51

F: Rest Station
Arrival Timestamps: 2021/1/6 17:58
Departure Timestamps: 2021/1/7 08:19

Fig. 13: A Case of Transportation Monitoring

6 Conclusion

We study the issue of transport stay hotspot recognizing for bulk commodity in the paper, and put forward a Multi-view Context awareness based transport Stay hotspot Recognization framework, called *MCSR*. Aiming at the issue of low precision of *Tshots*' location identification and functional tag annotation, we separately present a multi-view clustering based stay area merging strategy and an ensemble learning based annotation model embedded with a time-interval-aware self-attention network. Experimental results on a large scale real steel logistics dataset demonstrate that *MCSR* outperforms the state-of-the-art methods. In our future work, we will apply our transport stay hotspot recognization framework to more logistics scenarios to verify its rationality and validity.

References

[1] Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y.: Mining interesting locations and travel sequences from gps trajectories. In: WWW, pp. 791–800 (2009)

[2] Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y.: Understanding mobility based on gps data. In: UbiComp, pp. 312–321 (2008)

[3] Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y.: Mining correlation between locations using human location history. In: SIGSPATIAL, pp. 472–475 (2009)

[4] Ruan, S., Long, C., Yang, X., He, T., Li, R., Bao, J., Chen, Y., Wu, S., Cui, J., Zheng, Y.: Discovering actual delivery locations from mis-annotated couriers' trajectories. In: ICDE, pp. 3241–3253 (2022)

[5] Ruan, S., Xiong, Z., Long, C., Chen, Y., Bao, J., He, T., Li, R., Wu, S., Jiang, Z., Zheng, Y.: Doing in one go: Delivery time inference based on couriers' trajectories. In: SIGKDD, pp. 2813–2821 (2020)

[6] Zhu, Z., Ren, H., Ruan, S., Han, B., Bao, J., Li, R., Li, Y., Zheng, Y.: Icfinder: A ubiquitous approach to detecting illegal hazardous chemical facilities with truck trajectories. In: SIGSPATIAL, pp. 37–40 (2021)

[7] Ye, M., Shou, D., Lee, W.-C., Yin, P., Janowicz, K.: On the semantic annotation of places in location-based social networks. In: SIGKDD, pp. 520–528 (2011)

[8] Krumm, J., Rouhana, D.: Placer: semantic place labels from diary data. In: UbiComp, pp. 163–172 (2013)

[9] Krumm, J., Rouhana, D., Chang, M.-W.: Placer++: Semantic place labels beyond the visit. In: PerCom, pp. 11–19 (2015)

[10] Wu, T., Zhu, K., Mao, J., Yang, M., Zhou, A.: Tdcm: Transport destination calibrating based on multi-task learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 276–292 (2023)

[11] Hu, Y., Ruan, S., Ni, Y., He, H., Bao, J., Li, R., Zheng, Y.: Salon: a universal stay point-based location analysis platform. In: SIGSPATIAL, pp. 407–410 (2021)

[12] Pu, M., Mao, J., Du, Y., Shen, Y., Jin, C.: Road intersection detection based on direction ratio statistics analysis. In: MDM, pp. 288–297 (2019)

[13] Zhao, L., Mao, J., Pu, M., Liu, G., Jin, C., Qian, W., Zhou, A., Wen, X., Hu, R., Chai, H.: Automatic calibration of road intersection topology using trajectories. In: ICDE, pp. 1633–1644 (2020)

[14] He, T., Yin, H., Chen, Z., Zhou, X., Sadiq, S., Luo, B.: A spatial-temporal topic model for the semantic annotation of pois in lbsns. TIST **8**(1), 1–24 (2016)

[15] Hegde, V., Parreira, J.X., Hauswirth, M.: Semantic tagging of places based on user interest profiles from online social networks. In: European Conference on Information Retrieval, pp. 218–229 (2013)

[16] Wang, Y., Qin, Z., Pang, J., Zhang, Y., Xin, J.: Semantic annotation for places in lbsn through graph embedding. In: CIKM, pp. 2343–2346 (2017)

[17] Zhou, J., Gou, S., Hu, R., Zhang, D., Xu, J., Jiang, A., Li, Y., Xiong, H.: A collaborative learning framework to tag refinement for points of interest. In: SIGKDD, pp. 1752–1761 (2019)

[18] Dubey, M., Srijith, P., Desarkar, M.S.: Hap-sap: Semantic annotation in lbsns using latent spatio-temporal hawkes process. In: SIGSPATIAL, pp. 377–380 (2020)

[19] Zhou, M., Zhou, J., Fu, Y., Ren, Z., Wang, X., Xiong, H.: Description generation for points of interest. In: ICDE, pp. 2213–2218 (2021)

[20] Wang, B., Mezlini, A.M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., Goldenberg, A.: Similarity network fusion for aggregating data types on a genomic scale. Nature methods **11**(3), 333–337 (2014)

[21] Huang, Z., Zhou, J.T., Peng, X., Zhang, C., Zhu, H., Lv, J.: Multi-view spectral clustering network. In: IJCAI, vol. 2, p. 4 (2019)

[22] Wang, H., Yang, Y., Liu, B.: Gmc: Graph-based multi-view clustering. IEEE Transactions on Knowledge and Data Engineering **32**(6), 1116–1129 (2019)

[23] Li, L., Li, X., Li, Z., Zeng, D.D., Scherer, W.T.: A bibliographic analysis of the ieee transactions on intelligent transportation systems literature. IEEE Transactions on Intelligent Transportation Systems **11**(2), 251–255 (2010)

[24] Eddy, W.F.: A new convex hull algorithm for planar sets. TOMS **3**(4), 398–403 (1977)

[25] Jiang, J., Pan, D., Ren, H., Jiang, X., Li, C., Wang, J.: Self-supervised trajectory representation learning with temporal regularities and travel semantics. In: ICDE, pp. 843–855 (2023)

[26] Fernandes, E.R., Carvalho, A.C., Yao, X.: Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data. IEEE Transactions on Knowledge and Data Engineering **32**(6), 1104–1115 (2019)

[27] Hirsch, V., Reimann, P., Mitschang, B.: Exploiting domain knowledge to address multi-class imbalance and a heterogeneous feature space in classification tasks for manufacturing data. VLDB **13**(12), 3258–3271 (2020)

[28] Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018)

[29] Ying, L., Ke, Y.: Credit fraud detection for extremely imbalanced data based on ensembled deep learning. Journal of Computer Research and Development **58**(3), 539 (2021)

[30] Liu, Z., Tang, D., Cai, Y., Wang, R., Chen, F.: A hybrid method based on ensemble welm for handling multi class imbalance in cancer microarray data. Neurocomputing **266**, 641–650 (2017)

[31] Taherkhani, A., Cosma, G., McGinnity, T.M.: Adaboost-cnn: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. Neurocomputing **404**, 351–366 (2020)

[32] Ksieniewicz, P., Woźniak, M.: Dealing with the task of imbalanced, multidimensional data classification using ensembles of exposers. In: First International Workshop on Learning with Imbalanced Domains: Theory and Applications, pp. 164–175 (2017)

[33] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)

[34] Opitz, D.W.: Feature selection for ensembles. AAAI/IAAI **379**, 384 (1999)

[35] Chandra, A., Yao, X.: Ensemble learning using multi-objective evolutionary algorithms. Journal of Mathematical Modelling and Algorithms **5**(4), 417–445 (2006)

[36] Fang, H., Wang, Q., Tu, Y.-C., Horstemeyer, M.F.: An efficient non-dominated sorting method for evolutionary algorithms. Evolutionary computation **16**(3), 355–384 (2008)

[37] Poli, R., Langdon, W.B.: Genetic programming with one-point crossover. In: Soft Computing in Engineering Design and Manufacturing, pp. 180–189 (1998)

[38] Huang, Y., Xiao, Z., Yu, X., Wang, D., Havyarimana, V., Bai, J.: Road network construction with complex intersections based on sparsely sampled private car trajectory data. TKDD **13**(3), 1–28 (2019)

[39] Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Machine learning **45**(2), 171–186 (2001)

Declarations

Ethical Approval Not applicable.

Funding This study is supported by the Foundation of National Natural Science Foundation of China (Nos.62072180, U1911203).

Availability of data and materials Not applicable.