

Anomalous Trajectory Detection between Regions of Interest Based on ANPR System

Gao Ying, Nie Yiwen, Yang Wei, Xu Hongli, and Huang Liusheng

¹ University of Science and Technology of China, China

² {sa516067,nyw2016}@mail.ustc.edu.cn,
{qubit,xuhongli,lshuang}@ustc.edu.cn

Abstract. With the popularization of automobiles, more and more algorithms have been proposed in the last few years for the anomalous trajectory detection. However, existing approaches, in general, deal only with the data generated by GPS devices, which need a great deal of pre-processing works. Moreover, without the consideration of region's local characteristics, those approaches always put all trajectories even though with different source and destination regions together. Therefore, in this paper, we devise a novel framework for anomalous trajectory detection between regions of interest by utilizing the data captured by Automatic Number-Plate Recognition(ANPR) system. Our framework consists of three phases: abstraction, detection, classification, which is specially engineered to exploit both spatial and temporal features. In addition, extensive experiments have been conducted on a large-scale real-world datasets and the results show that our framework can work effectively.

Keywords: Anomalous trajectory · Regions of interest · ANPR system.

1 Introduction

It has been well known that “one person’s noise could be another person’s signal.” Indeed, for some applications, the rare is more attractive than the usual. For example, when mining vehicle trajectory data, we may pay more attention to the anomalous trajectory since it is helpful to the urban transportation analysis.

Anomalous trajectory is an observation that deviates so much from other observations as to arise suspicious that it may be generated by a different mechanism. Analyzing such type of movement between regions of interest is beneficial for us to understand the road congestion, reveal the best or worst path, locate the main undertaker when traffic accidents happen and so on.

Existing trajectory-based data mining techniques mainly exploit the geolocation information provided by on-board GPS devices. [1] takes advantage of real-time GPS traffic data to evaluate congestion; [2] makes use of GPS positioning information to detect vehicles’ speeding behaviors; [21] utilizes personal GPS walking trajectory to mine frequent route patterns. Exploiting GPS data to detect anomalous trajectories has a good performance. However, there are considerable overhead in installing GPS devices and collecting data via networks.

In this paper, we devise a novel framework for anomalous trajectory detection between regions of interest based on the data captured by ANPR system. In an ANPR system, a large number of video cameras are deployed at various locations of an area to capture and automatically recognize their license plate numbers of passing by vehicles. Each of location is often referred to as an ANPR gateway. And the trajectory of a vehicle is the concatenation of a sequence of gateways.

Compared to existing techniques that make use of GPS data, exploiting ANPR records in anomalous trajectory detection has the following advantages: high accuracy in vehicle classification, low costs of system deployment and maintenance, better coverage by monitoring vehicles and so on.

In summary, we make the following contributions in contrast to existing approaches:

1. We introduce ANPR system that not only can constantly and accurately reveal the road traffic but also almost does not need additional pre-processing works.
2. We devise a novel framework to detect anomalous trajectory between regions of interest. Specifically, we take the road distribution and road congestion into consideration.
3. Finally, using the real monitoring records, we demonstrate our devised framework can detect the anomalous trajectories correctly and effectively.

The rest of this paper is organized as follows. Section II presents the related works. Section III provides the problem statement. Section IV gives our specific anomalous trajectory detection algorithms. Section V describes the results of experimental evaluation. Finally, the concluding remarks are drawn in Section VI.

2 Related Work

Here, we review some related and representative works. And this section can be categorized into two parts. The first part will revolve around outlier detection algorithms, whereas the second part will concentrate on the existing anonymous trajectory detection algorithms.

2.1 Outlier Detection Algorithms

A great deal of outlier detection algorithms have been developed for multi-dimensional points. These algorithms can be mainly divided into two classes: distance-based and density-based.

1. ***Distance-based method:*** This method is originally proposed in [15–17, 7]. “An object O in a dataset T is a $DB(p,D)$ -outlier if at least fraction p of the objects in T lies greater than distance D from O .” This method relies deeply on the global distribution of the given dataset. So if the distribution conforms to or approximately conforms to uniform distribution, this algorithm can perform perfectly. However, it encounters difficulties when analyzing the dataset with various densities.

2. **Density-based method:** This method is proposed in [18,19]. A point is classified into an outlier if the *local outlier factor(LOF)* value is greater than a given threshold. Here, each point's LOF value depends on the local densities of its neighborhoods. Clearly, the LOF method dose not suffer from the problem above. However, the computation of LOF values require a great batch of k-nearest neighbor queries, and thus, can be computationally expensive.

2.2 Anomalous Trajectory Detection Algorithms

In recent years, more and more researchers have paid their attention to anomalous trajectory detection[3, 6, 14, 5]:

Fontes V and De Alencar L[3] give a novel definition of standard trajectory in their paper, and propose that if there is at least one standard path that has enough neighborhoods nearby, then a potential anomalous trajectory that does not belong to standard group would be regarded to perform a detour, and is classified into anomalous. This rather simplistic approach even though can find out all anomalous trajectories, quantities of normal trajectories are incorrectly classified.

JG Lee, J Han and X Li[6] propose a novel partition-and-detect framework. In their paper, they claim that even though some partitions of a trajectory show an unusual behavior, these differences may be averaged out over the whole trajectory. So, they recommend to split a trajectory into various partitions(at equal intervals), and a hybrid of distance- and density-based approaches are used to classify each partition as anomalous or not, as long as one of the partitions is classified into anomalous, the whole trajectory is considered as anomalous. However, solely using distance and density can fail to correctly classify some trajectories as anomalous.

Xiaolei Li[14] present an anomalous trajectory detection algorithm based on classification. In their algorithm, they first extract some common patterns named motifs from trajectories. And then they transform the set of motifs into a feature vector which will be fed into a classifier. Finally, through their trained classifier a trajectory is classified into either "normal" or "anomalous". Obviously, their algorithm depends deeply on training. However, in a real world, it is not always easy to obtain a good training set. Notice that our algorithm does not require such training.

Due to the inherent drawbacks of the GPS devices, some researchers have turned their attention to the ANPR system. A Homayounfar [20] apply data clustering techniques to extract relevant traffic patterns from the ANPR data to detect and identify unusual patterns and irregular behavior of multi-vehicle convoy activities. Yuyan Sun[4] propose a new anomaly detection scheme that exploits vehicle trajectory data collected from ANPR system. Their scheme is capable of detecting vehicles with the behavior of wandering round and unusual activity at specific time. However, these methods are too one-side, and there is no effective and comprehensive method to detect anomalous trajectory.

3 Problem Statement

In this section, we give several basic definitions and the formal problem statement. Before that, we make a brief synopsis of our dataset.

As mentioned before, our dataset were collected from ANPR system. By processing the ANPR data, we could get each vehicle's historical ANPR records. Each ANPR record includes the captured time, the gateway id of the capturing camera, and the license of the captured vehicle[4]. And by asking Traffic Police Bureau for help, we can obtain the latitude and longitude of every on-line gateway id.

Definition 1. (TRAJECTORY) A *trajectory* consists of a sequence of passing by points $[p_1, p_2, \dots, p_n]$, where each point is composed of the captured time, the latitude and the longitude of the surveillance camera.

Definition 2. (CANDIDATE TRAJECTORY) Let SRC, DEST be the source region and the destination region of interest and $t = [p_1, p_2, \dots, p_n]$ is a trajectory. t becomes a *candidate trajectory* if and only if the source region $P_1 = \text{SRC}$ and the destination region $P_n = \text{DEST}$.

Candidate group is a set of candidate trajectories.

Definition 3. (NEIGHBORHOOD) Let t be a candidate trajectory, the neighborhoods of t can be collected by the following formula:

$$N(t, \text{maxDist}) = \{c_i \mid c_i \text{ is a candidate and } \text{dist}(t, c_i) \leq \text{maxDist}\}.$$

where $\text{dist}(t, c_i)$ can be calculated by the use of *Algorithm 2*, and the maxDist means maximum distance, it is a predefined threshold.

Definition 4. (STANDARD TRAJECTORY) Let t be a candidate trajectory, t is a *standard trajectory* if and only if $|N(t, \text{maxDist})| \geq \text{minSup}$, where minSup means minimum support, it is also a predefined threshold.

Standard group is a set of standard trajectories.

Definition 5. (ANOMALOUS TRAJECTORY) A candidate trajectory will be classified into anomalous if it satisfies both of the following requirements:

1. the similarity between the candidate trajectory and the standard group is less than a given threshold S;
2. the difference between the candidate trajectory and the standard group is more than a given threshold D;

PROBLEM STATEMENT: Given a set of trajectories $T = \{t_1, t_2, \dots, t_n\}$, a fixed S-D pair (S, D) and a candidate trajectory $t = [p_1, p_2, \dots, p_n]$ moving from S to D. We are aimed to verify whether t is anomalous with respect to T. Furthermore, we would like to reveal the anomalous score that will be used to arrange the processing priority.

4 Anomalous Trajectory Detection Framework

In this section, we introduce our devised anomalous trajectory detection framework in details. This framework is mainly divided into three phases: abstraction, detection, classification.

4.1 Abstraction

The abstraction is aimed to abstract the candidate group and the standard group between regions of interest from a large number of unorganized ANPR records.

The first step of which is to synthetic a vehicle's trajectory. By the hand of ANPR system, we can synthetic a trajectory which is composed of the vehicle's captured records in a whole day. However, analyzing the entire trajectory of a vehicle may not be able to extract enough features. Thus, we decide to partition the whole trajectory into a set of sub trajectories based on the time interval between records. Each sub trajectory indicates an individual short-term driving trip. And in a sub trajectory, the time interval between records must be less than practical threshold *Duration*.

The second step of which is to abstract the candidate group and the standard group. By the use of the definitions presented at *Def.2* and *Def.4*, we can abstract them quickly. However, we may run into a bad situation when we apply the method to a desert region(the desert means the region is desolate and there are so little passing by vehicles). In a desert region, there may be not enough vehicle's monitoring trajectories for us to abstract standard group. In this situation, we can find out 5 most frequently used paths to compose our standard group.

4.2 Detection

The detection is intended to calculate the similarity and difference between the candidate and the standard group. In this section, we propose *adjusting weight longest common subsequence(AWLCS)* to calculate the similarity and *adjusting weight dynamic time warping(AWDTW)* to calculate the difference.

Adjusting Longest Common Weighted Subsequence In the beginning, we introduce the famous NP-hard problem LCS:

Problem 1. The string Longest Common Subsequence(LCS) Problem:

INPUT: *Two trajectories t_1, t_2 of length n, m ;*

OUTPUT: *The length of the longest subsequence common to both strings.*

For example, for $t_1=[p_1, p_2, p_3, p_4, p_4, p_1, p_2, p_5, p_6]$ and $t_2=[p_5, p_6, p_2, p_1, p_4, p_5, p_1, p_1, p_2]$, $LCS(t_1, t_2)$ is 4, where a possible such subsequence is $[p_1, p_4, p_1, p_2]$.

Using LCS algorithm to calculate the similarity between two trajectories gives good results when the captured cameras are deployed at approximately equidistance. But if not, a problem arises. The problem is the following: some cameras are adjacent with each other, while some cameras are remote with each

other, just like the situation depicted in *Fig. 1*. Now when we apply LCS to calculate the similarity between two trajectories, all cameras are deemed as equally important(in fact, the remote cameras play a more important role than the adjacent cameras), which neglects the road distribution definitely leading to a bad result.

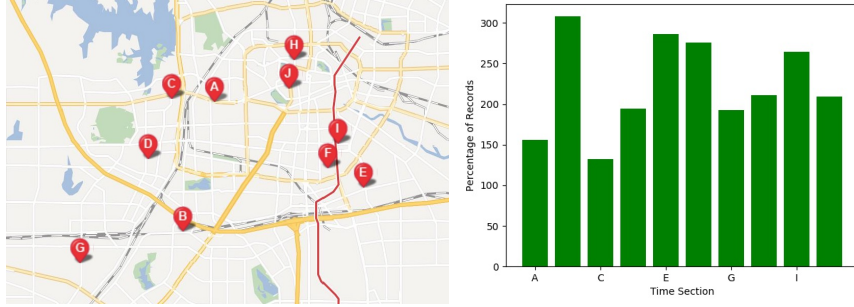


Fig. 1. non-equidistant cameras

Fig. 2. traffic volumes of captured cameras

One good way to solve this problem is to allocate different weights to different captured cameras: smaller weights to cameras that are located in dense area and bigger weights to the cameras that are located in sparse area. In there, we abstract the cameras into points. Weight of point i (w_i) can be calculated, for instance, by using the following equation:

$$w_i = \frac{c_i}{\sum_{k=0}^{k=n-1} c_k}, \quad (1) \quad \text{where}$$

$$c_i = \begin{cases} \frac{\text{dist}(p_2, p_1)}{\text{equidistant}}, & i = 0 \\ \frac{\text{dist}(p_{i+1}, p_i) + \text{dist}(p_i, p_{i-1})}{2 * \text{equidistant}}, & 1 < i < n - 1 \\ \frac{\text{dist}(p_n, p_{n-1})}{\text{equidistant}}, & i = n - 1 \end{cases} \quad (1)$$

The variable *equidistant* tells the distance interval on the condition that the points of a trajectory are distributed at equidistance:

$$\text{equidistant} = \frac{\text{dist}(p_n, p_1)}{n-1} \quad (2)$$

And coefficient C_i tells how far the neighbors of point p_i are located compared with a case where the points are distributed at approximately equidistant. Note that in the case of $0 < i < n - 1$, the points have two neighbors, while in the case of $i = 0$ and $i = n - 1$, the points only have one neighbor.

Now, we can present *AWLCS* in *Algorithm 1*.

By *Algorithm 1*, we can obtain the similarity measure between the candidate and one standard. As for the similarity between the candidate and the standards is the maximum between the candidate and the standard in group.

Algorithm 1: Adjusting Weight Longest Common Subsequence

Input: A, B
Output: *the longest common weighted subsequence*

```

1  $m := \text{length}(A)$ ;
2  $n := \text{length}(B)$ ;
3 if  $m == 0$  or  $n == 0$  then
4    $\lfloor$  return 0;
5 else if  $A[m] == B[n]$  then
6    $\lfloor$  return  $\text{weight}(B[n]) + \text{AWLCS}(\text{Head}(A), \text{Head}(B))$ ;
7 else
8    $\lfloor$  return  $\max(\text{AWLCS}(\text{Head}(A), B),$ 
9      $\text{AWLCS}(A, \text{Head}(B)))$ ;

```

Adjusting Weight Dynamic Time Warping We now discuss the problem of computing the difference between a candidate and the standard group using *AWDTW*.

The simplest way of calculating dynamic time warping is given by [13] using dynamic programming. This method is mainly divided into two steps. The first step is to evaluate the distance matrix of two trajectories. And the second step is to find the shortest path moving from the lower left corner $\text{DTW}[0,0]$ to the upper right corner $\text{DTW}[n,m]$. The pseudo-code is presented in *Algorithm 2*.

For point $[i, j]$, it only can be arrived at insertion(previous point is $[i-1, j]$), or deletion(previous point is $[i, j-1]$), or match(previous point is $[i-1, j-1]$). So $[i, j]$ must choose one of the three distance extensions to pass through point $[i, j]$, at this time, the cumulative distance is calculated as (line 12,13,14).

When we apply DTW to calculate the difference between two trajectories, we take the road congestion into consideration. It's obvious that the traffics among different roads is different, some differences even are much huge, just like the situation depicted in *Fig. 2*. So some experienced drivers may choose an unusual trajectory that though may deviates from the standard group, to avoid congestion.

Therefore, similar to AWLCS, the AWDTW also allocate a weight to the captured camera. However, the definition of weight in AWDTW is much different from that in ALCWS. Under this circumstance, we can calculate weight w_i as the ratio of average traffic volume to the traffic volume of p_i .

$$\text{weight}(p_i) = \frac{\sum_{p_k \in \phi} \text{Vol}(p_k) / |\phi|}{\text{Vol}(p_i)} \quad (3)$$

where $\text{Vol}(p_k)$ is the traffic volume of p_k at a certain duration and ϕ is the collection of all points.

By this calculation, we will obtain a low value $\text{weight}(p_i)$ when the point p_i 's congestion is heavier than the average, and a high value $\text{weight}(p_i)$ when the point p_i 's congestion is lighter than the average. The bigger of the value $\text{weight}(p_i)$, the higher of the chance that p_i is chosen .

Algorithm 2: Dynamic Time Warping

Input: A, B
Output: $DTW[n, m]$

```

1 m := length(A);
2 n := length(B);
3 DTW := array[0...n, 0...m];
4 for i = 1; i ≤ n; i ++ do
5   DTW[i, 0] := infinity;
6 for j = 1; j ≤ m; j ++ do
7   DTW[0, j] := infinity;
8 DTW[0,0] = 0;
9 for i = 1; i ≤ n; i ++ do
10  for j = 1; j ≤ m; j ++ do
11    cost := distance(s[i], t[j]);
12    DTW[i, j] := cost + minimum(DTW[i-1, j ],
13                               DTW[i , j-1],
14                               DTW[i-1, j-1]);
15 return DTW[n, m]
```

When we compute the weight of a point p_k , $\text{Vol}(p_k)$ may be zero, which will bring about a serious impact on the following computation. So, we add an initial value 1 to every $\text{Vol}(p_k)$.

After defining the weight value, when we compute the distance between p_i in standard trajectory and p_j in the candidate trajectory, the distance is multiplied by the weight(p_j). After the adjustment, the distance is decreased in a congested region due to a low weight(p_j) value(< 1.0), but it is increased in a uncongested region due to a high weight(p_j) value(> 1.0).

Before we give the difference between the standard group and the candidate trajectory, we first introduce the inter-group distance and intra-group distance:

INTER-GRPUP DISTANCE: The inter-group distance ω is the distance between the standard group and the candidate trajectory, which is equal to the minimum between the candidate and the standard trajectory in standard group.

INTRA-GROUP DISTANCE: The intra-group distance u is the maximum distance of any two trajectories in standard group.

In order to calculate the intra-group distance u , the size of the standard group must be greater than or equal to 2. So when there is only 1 element in group, this method has lost efficacy. Under this circumstance, we randomly select 5 candidate trajectories to form our standards, at this time, u is equal to the minimum between any two standard trajectories in standard group.

After acquiring the inter-group distance and the intra-group distance, the *difference* between the candidate and the standard group can be calculated as following:

$$difference = \frac{|\omega - u|}{u} \quad (4)$$

where the ω tells us the distance between the the candidate trajectory and the standard group, and the u can be regarded as the distance between a standard trajectory and the standard group, thus the $|\omega - u|$ reveals that how far away the candidate trajectory in contrast to a standard trajectory. However, we can not directly use the $|\omega - u|$ to stand for difference, because there is a great difference for different source region and the destination region. By dividing the u , it can neglect this effect.

4.3 Classification

The classification is designed to classify the candidate trajectory into anomalous or normal according to the similarity and difference calculated in previous stage. The concrete classification method is presented in *Def. 5*.

Once a candidate is classified into anomalous, some actions should be taken at once. But, if a great deal of candidate trajectories are classified into anomalous at the same time, what's the processing sequences? Obviously, The bigger anonymity, the higher processing priority. Thus, we propose anomalous score to show its level of anonymity, whose computational formula is presented in the following:

$$\Delta(s, d) = \lfloor e^{\lambda(S-s)} + e^{\lambda(d-D)} \rfloor \quad (5)$$

Here, λ is a temperature parameter, s and d are the calculated similarity and difference, S and D are the aforementioned similarity threshold and difference threshold. For our experiments, we choose $\lambda = 500$. According to the computation formula , we can conclude that the bigger the similarity s , the smaller the anomalous score; the bigger the difference d , the bigger the anomalous score.

5 Experimental Evaluation

Here, we provide an empirical evaluation and analysis of our devised framework. All the experiments are run in Python 3.5 on Mac OS.

5.1 Experimental Dataset

Our dataset were collected from the ANPR system deployed at Hefei between August 15, 2017 and August 23, 2017. The total number of ANPR records is close to 100 million, which includes about 10 million trajectories. However, there is still a lack of anomalous trajectories, so we simulate the illegal drivers' escape inspection behaviors and taxi drivers' detour behaviors in real environment.

5.2 Evaluation Criteria

A classified candidate trajectory will fall into one of the following four scenarios:

1. True positive(TP): an anomalous trajectory is correctly labeled to anomalous;

2. False positive(FP): a non-anomalous trajectory is incorrectly labeled to anomalous;
3. False negative(FN): an anomalous trajectory is incorrectly labeled to non-anomalous;
4. True negative(TN): a non-anomalous trajectory is correctly labeled to non-anomalous.

According to the number of TP, FP, FN, TN, we can get the *Precision* and the *Recall*:

Precision: The *precision* concentrates on this problem: of all trajectories where we labeled to anomalous, what fraction actually be correctly labeled.

$$Precision = \frac{TP}{TP+FP}$$

Recall: The *recall* concentrates on this problem: of all trajectories that actually is anomalous, what fraction did we correctly labeled to anomalous.

$$Recall = \frac{TP}{TP+FN}$$

Obviously, the bigger of *precision and recall*, the better performance of this binary classification. However, you can't have your cake and eat it too. Thus, taking these two indicator into consideration, we choose *F1-measure* to evaluate the performance of this binary classification.

$$F1\text{-measure} = \frac{2*Precision*Recall}{Precision+Recall}$$

5.3 Parameters Setting

As in many other data mining algorithm, the parameters setting is essential for the final experiment results. In our framework, we need to set the following five parameters: *duration threshold*, *maxDist*, *minSup*, *similarity threshold*, *difference threshold*.

Duration Threshold: In the actual drive test at Hefei, the duration that a vehicle passes through two adjacent captured point is no more than 30 minutes even though during rush hours. Besides, we have calculated and analysis the ANPR records, its distribution is presented at *Fig. 3*, the proportion of the duration that less than 30 minutes is up to 74.6%. Thus, we set the *duration threshold = 30 min*.

maxDist and minSup: It is obvious that with the increase of *maxDist*, more candidate trajectories would be included in standard group; however, with the increase of *minSup*, less candidate trajectories would be included in standard group. And the bigger of the size of standard group, the less of the chance of a candidate trajectory classified to anomalous. So it is important to investigate their effects on performance. In *Fig.4* we plot the F1 value varies from *maxDist*, and In *Fig.5*, we plot the F1 value varies from *minSup*. From these two pictures, we can conclude that the *maxDist* should not be set any lower than 90 and the *minSup* should not be set any higher than 20.

Similarity Threshold and Difference Threshold: Since similarity and difference is the threshold for determining anomalousness, it is important to investigate its effect on the performance of our devised framework. We study the effect on performance when similarity ranges from 0.0 to 1.0 and difference

ranges from 1.0 to 3.0. In *Fig.6*, we plot the F1 value for different values of similarity and In *Fig. 7*, we plot the F1 value for different values of difference. We can see that similarity should be set to 0.5 and difference should be set to 2.0 since less than or more than them the performance would significantly decrease.

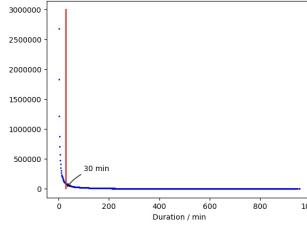


Fig. 3. Duration

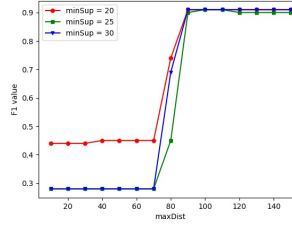


Fig. 4. maxDist

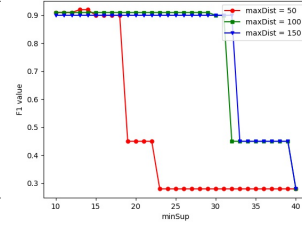


Fig. 5. minSup

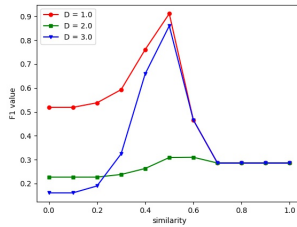


Fig. 6. Similarity

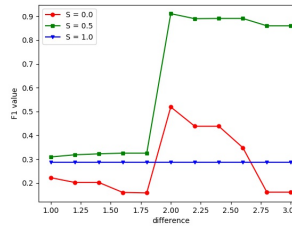


Fig. 7. Difference

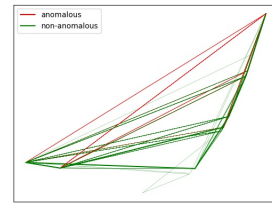


Fig. 8. Performance

5.4 Evaluation

Fig.8 shows the result between two interest regions. We observe that many anomalous trajectories are detected. Some of the anomalous trajectories deviate so much from normal group, which are mainly caused by illegal drivers that choose an unusual path to escape polices' inspection; And some trajectories even though are similar to most of the normal trajectories, they are still be classified into anomalous, which are mainly caused by taxi drivers' and dripping drivers' detour behaviors.

Next, we evaluate the superiority of our proposed framework.

We compare its performance with two existing anomalous detection approaches: discovering trajectory outlier between regions of interest(ROF) presented in [3] and trajectory outlier detection: a partition-and-detect framework(PAD)

Algorithms	Precision	Recall	F1 value
ROF	0.425	1.0	0.597
PAD	0.724	0.868	0.789
Our Proposal	0.927	0.895	0.911

Table 1. Compare with ROF and PAD

presented in [6]. We show the experiment results in *Table. 1*. We can see that ROF has the best recall, but worst precision, resulting in a great waste of human, material and financial resources. As for PAD, it ignores temporal information in trajectory which is of great importance, so the running results are still not very good.

6 Conclusion

In this paper, we devise a novel framework for anomalous trajectory detection between regions of interest based on the data captured by Automatic Number-Plate Recognition (ANPR) system. Taking both spatial and temporal features into consideration, we propose two new algorithms AWLCS and AWDTW. A large number of experiments manifest that our framework significantly outperforms existing schemes.

References

1. Bacon J, Bejan A I, Beresford A R, et al. Using Real-Time Road Traffic Data to Evaluate Congestion[M]// Dependable and Historic Computing. Springer Berlin Heidelberg, 2011:93-117.
2. Mohamad I, Ali M A M, Ismail M. Abnormal driving detection using real time Global Positioning System data[C]// IEEE International Conference on Space Science and Communication. IEEE, 2011:1-6.
3. Fontes V, De Alencar L, Renso C, Bogorny V: Discovering trajectory outliers between regions of interest, Proceedings of the Brazilian Symposium on Geoinformatics (2013) pp. 49-60 Published by National Institute for Space Research, INPE
4. Sun Y, Zhu H, Liao Y, et al. Vehicle Anomaly Detection Based on Trajectory Data of ANPR System[C]// Global Communications Conference. IEEE, 2016:1-6.
5. Chen C, Zhang D, Castro P S, et al. iBOAT: Isolation-Based Online Anomalous Trajectory Detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(2):806-818.
6. Lee J G, Han J, Li X. Trajectory Outlier Detection: A Partition-and-Detect Framework[J]. 2008:140-149.
7. Papadimitriou S, Kitagawa H, Gibbons P B, et al. LOCI: fast outlier detection using the local correlation integral[C]// International Conference on Data Engineering, 2003. Proceedings. IEEE, 2002:315-326.
8. Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets[C]// ACM SIGMOD International Conference on Management of Data. ACM, 2000:427-438.

9. Bunke H. On a relation between graph edit distance and maximum common sub-graph[J]. *Pattern Recognition Letters*, 1997, 18(9):689-694.
10. Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories[C]// *International Conference on Data Engineering*, 2002. *Proceedings. IEEE*, 2002:673.
11. Berndt D J. Using dynamic time warping to find patterns in time series[C]// *Kdd Workshop*. 1994:359-370.
12. Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space[J]. *Intelligent Data Analysis*, 2007, 11(5):561-580.
13. Yi B K, Jagadish H V, Faloutsos C. Efficient Retrieval of Similar Time Sequences Under Time Warping[C]// *International Conference on Data Engineering*, 1998. *Proceedings. IEEE*, 2002:201-208.
14. Li X, Han J, Kim S, et al. ROAM: Rule- and Motif-Based Anomaly Detection in Massive Moving Object Data Sets[C]// *Siam International Conference on Data Mining*, April 26-28, 2007, Minneapolis, Minnesota, Usa. *DBLP*, 2007.
15. Knorr E M, Ng R T. Algorithms for Mining Distance-Based Outliers in Large Datasets[C]// *International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc. 1998:392-403.
16. Knorr E M, Ng R T, Tucakov V. Distance-based outliers: algorithms and applications[J]. *Vldb Journal*, 2000, 8(3-4):237-253.
17. Knorr E M, Ng R T. Finding Intensional Knowledge of Distance-Based Outliers[J]. *Vldb*, 1999:211-222.
18. Breunig M M, Kriegel H P, Ng R T. LOF: identifying density-based local outliers[C]// *ACM SIGMOD International Conference on Management of Data*. *ACM*, 2000:93-104.
19. Jin W, Tung A K H, Han J. Mining top-n local outliers in large databases[C]// 2001:293-298.
20. Homayounfar A, Ho A T S, Zhu N, et al. Multi-vehicle convoy analysis based on ANPR data[C]// *International Conference on Imaging for Crime Detection and Prevention*. *IET*, 2012:1-5.
21. Fu Z, Tian Z, Xu Y, et al. Mining frequent route patterns based on personal trajectory abstraction[J]. *IEEE Access*, 2017, PP(99):1-1.