

# Trajectory Anomaly Detection with By-Design Complementary Detectors

Shurui Cao\*

Leman Akoglu\*

## Abstract

Trajectory anomaly detection is critical across a wide range of applications, from traffic control, and wildlife conservation, to public transportation optimization. However, detecting anomalies in trajectory data is challenging due to the diverse nature of anomalies. In this paper, we propose CETRAJAD, an ensemble method for trajectory anomaly detection that integrates complementary detectors, each targeting different aspects of trajectory anomalies. Our approach leverages three types of trajectory embeddings—Route, Speed, and Shape—that vary in their sensitivity to length, direction, shape, and speed, enabling the detection of diverse anomaly types. We combine detectors from both the embedding and input spaces and show how their complementary nature improves anomaly detection performance. Through theoretical analysis, we demonstrate the conditions when the proposed ensemble design outperforms traditional ensemble methods. Experiments on multiple real-world datasets, containing both simulated and ground-truth anomalies, show that the proposed model consistently outperforms existing baselines.

**Keywords:** Trajectory Anomaly Detection, Outlier Ensemble, Trajectory Mining

## 1 Introduction

Given a large collection of trajectories, how can we find deviations and anomalies? Trajectory data, depicting latitude and longitude coordinates over time, captures how entities move. Those entities could vary from individuals and animals to trucks and airplanes, with a wide range of anomaly detection applications. For example, anomalies in airplane trajectories could be early signals of mechanical failure or other atypical scenarios onboard [13]. Delivery and logistics companies may monitor their trucks and drivers to spot deviations with potential implications for delays [25]. Pedestrian trajectories can signal inefficiencies in urban design [17].

Trajectory anomaly detection is nontrivial as what constitutes an anomaly is often under-specified. Our motivation stems from the many possible definitions and different types of anomalies that may exist in real-world

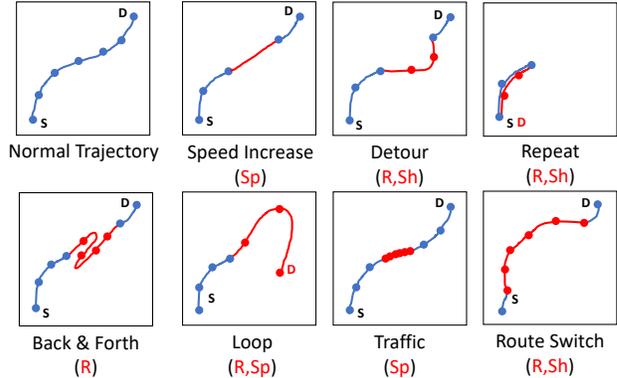


Figure 1: (best in color) Example anomalous trajectories with anomalies (highlighted in red) of various types involving route (R), shape (Sh), and/or speed (Sp).

data. Specifically, trajectory anomalies may arise from the route, the speed, and the interaction between the two. Figure 1 illustrates various examples, including (i) route anomaly, an individual taking a detour or a longer-than-usual trip; (ii) speed anomaly, making stop(s) along the typical route, e.g., due to traffic or to pick-up things; (iii) route-speed interaction anomaly, driving faster than would be safe on part of the route with hairpin turns, among others.

Though trajectory anomaly mining has been widely studied [14, 4], most existing methods focus on specific domains (e.g., autonomous driving [21, 10]) or specific entity trajectories (e.g., taxi trajectories [20, 8]), which limits their generalizability and the ability to accommodate the diversity of anomaly types present across different mobility datasets.

In this paper, we present a deep ensemble model named CETRAJAD for trajectory anomaly detection, which is composed of *complementary* detectors based on three different trajectory embeddings, specifically designed to capture varying (in)variances in terms of length, direction, and speed. The ensemble includes detectors that operate in both the embedding space (e.g., using the density of the learned embeddings) as well as the input space (e.g., using reconstruction loss to measure the deviation of original data). To the best of our knowledge, we are the first to underscore the variety in trajectory anomalies, and propose a

\*Heinz College of Information Systems, Carnegie Mellon University. {shuruic, lakoglu}@andrew.cmu.edu

model that can detect different types of anomalies through complementary detectors *by design*. The main contributions are summarized as follows.

- **Trajectory embedding with varying (in)variances:** We learn three different embeddings for trajectory data; namely, Route, Speed, and Shape, that exhibit varying (in)variances with respect to length, direction, and speed. This variation in representation serves as the foundation for detecting different types of trajectory anomalies.
- **Ensemble of complementary anomaly detectors:** We employ anomaly detectors both in the embedding and the input space, and demonstrate how they complement each other (i.e., detect different anomalies). As our detectors exhibit low correlation, as opposed to consensus, we design a new ensemble method to reach a composed ranking that is better than its parts.
- **Evaluation:** We evaluate CETRAJAD on multiple real-world datasets based on both simulated and ground truth anomalies and show that it outperforms existing baselines, especially when the data contains a mixture of different types of trajectory anomalies.

**Access and Reproducibility:** All our codebase for trajectory embedding and ensemble anomaly detection with complementary detectors is open-sourced at <https://github.com/ShuruiCao/comp-ensemble-ad>.

## 2 Problem and Preliminaries

We introduce related concepts and formulate the anomaly detection problem on trajectories.

**Trajectory.** A trajectory is a chronological sequence of GPS coordinates with timestamps, denoted as  $T = [p_1, p_2, p_3, \dots, p_n]$ , where  $p_i = (\text{lat}_i, \text{lon}_i, t_i)$ , and lat, lon and  $t$  represent the latitude, longitude, and timestep.

**Normalized Trajectory.** To improve flexibility and generalizability, we normalize each trajectory by subtracting the latitude and longitude of the source location from all subsequent locations. For a raw trajectory  $T$ , its normalized trajectory is  $T_{\text{normalized}} = [\Delta p_1, \Delta p_2, \Delta p_3, \dots, \Delta p_n]$ , where  $\Delta p_i = (\Delta \text{lat}_i, \Delta \text{lon}_i, t_i)$  represents the change in location from the source location:  $\Delta \text{lat}_i = \text{lat}_i - \text{lat}_1$  and  $\Delta \text{lon}_i = \text{lon}_i - \text{lon}_1$ .

**Anomalous Trajectory.** A trajectory is considered anomalous if its pattern significantly deviates from other trajectories. Previous works [12, 7, 24] often define an anomalous trajectory in the context of a specific source-destination (SD) pair: a trajectory is considered an outlier if it occurs infrequently and differs from other trajectories associated with that particular SD pair. In contrast, our approach takes a broader perspective,

considering a trajectory to be an outlier if it deviates from typical patterns at the city level, without restricting the analysis to specific SD pairs.

**Unsupervised Trajectory Anomaly Detection.** Given a trajectory  $T$ , we aim to learn an anomaly score function to identify whether  $T$  is anomalous, i.e. deviating significantly from common patterns observed in other trajectories, without relying on labeled data.

## 3 Complementary Detectors by Design

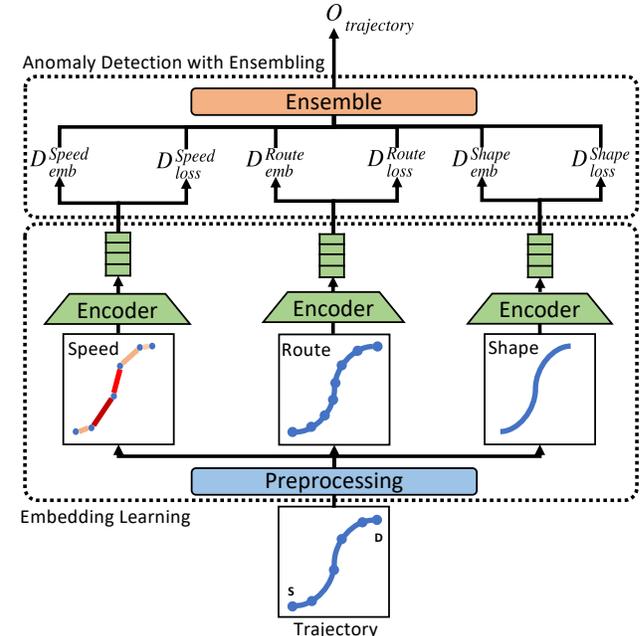


Figure 2: Proposed pipeline for CETRAJAD.

In this section, we introduce the proposed CETRAJAD. As illustrated in Figure 2, CETRAJAD comprises two main components:

- **Trajectory Embedding Learning** We design three different preprocessing methods to capture varying characteristics of trajectories in length, direction, speed, and shape, and then learn three types of embeddings that are complementary *by design*, as they capture varying (in)variances.
- **Anomaly Detection with Ensembling** We employ various detectors based on embeddings and design a complementary ensemble method to detect anomalous trajectories.

### 3.1 Complementarity by Semantic Embeddings

Complementarity means different embeddings capture semantically distinct aspects of the data. That is, properties captured by one embedding are not always captured by others, which as a result, induces low correlation between downstream detectors.

Table 1: Comparison of embedding types (S, R, Sh) and their invariances to different aspects of trajectories.

	(S)peed	(R)oute	(Sh)ape
Length	<b>Sensitive</b>	<b>Sensitive</b>	Invariant
Direction	Invariant	<b>Sensitive</b>	Invariant
Speed	<b>Sensitive</b>	Invariant	<b>Sensitive</b>
Start location	Invariant	Invariant	Invariant

In the real world, trajectory anomalies can arise in various forms, such as speed, route taken, direction, length, etc. To capture the diverse characteristics of trajectories, we design three distinct processing methods to obtain complementary embedding types: Speed, Route, and Shape embedding. Their characteristics are summarized in Table 1.

In the following, we describe the data transformations/preprocessing for each embedding. Each transformed trajectory data is fed into a Long Short-Term Memory (LSTM)[9] autoencoder for obtaining the respective embedding.

**3.1.1 Speed embedding** Speed embedding focuses on capturing dynamic changes in the trajectory by quantifying how the speed changes. The input is the normalized trajectory, and the speed is computed as the distance traveled between consecutive timesteps:  $v_i = \frac{d(\Delta p_i, \Delta p_{i-1})}{t_i - t_{i-1}}$  where  $d(\Delta p_i, \Delta p_{i-1})$  is the Haversine distance between consecutive locations  $\Delta p_i$  and  $\Delta p_{i-1}$ , and  $t_i$  and  $t_{i-1}$  are the consecutive timesteps. The output of the preprocessing is the sequence of speed values,  $[v_i, \dots, v_{n-1}]$ .

**3.1.2 Route embedding** Route embedding aims to obtain a speed-invariant representation and focuses on the spatial path taken by the trajectory. This is achieved by interpolating the normalized trajectory based on unit cumulative distance. The cumulative distance is computed as:  $S_i = \sum_{j=1}^i d(\Delta p_j, \Delta p_{j-1})$  where  $S_i$  is the cumulative distance up to the  $i$ -th point. We then interpolate the trajectory by resampling the trajectory at fixed intervals of cumulative distance  $\Delta S$  which is a parameter:  $\Delta p'_i = \text{interpolate}(\Delta p_i, S_i \bmod \Delta S = 0)$ . The output is the sequence of interpolated points,  $[\Delta p'_1, \dots, \Delta p'_n]$ . This ensures that the points in the trajectory are uniformly distributed in terms of the distance traveled, making the embedding invariant to the rate of movement. It still captures the directionality and length information of the trajectory.

**3.1.3 Shape embedding** Shape embedding captures the geometric shape of the trajectory and deempha-

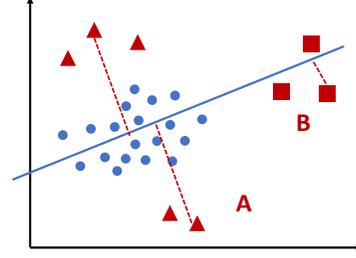


Figure 3: Complementarity between embedding space and input space. Blue points denote normal data and the blue line represents the embedding space. Red squares and triangles denote different types of anomalies.

sizes the length and directional information. We firstly normalize the coordinates and timesteps of the normalized trajectory by the total cumulative distance so all trajectories have the same cumulative distance  $S_n$ :  $\Delta p'_i = \left( \frac{\Delta \text{lat}_i}{S_n}, \frac{\Delta \text{lon}_i}{S_n}, \frac{t_i}{S_n} \right)$  We resample trajectory points at the frequency of 24 seconds by interpolating or down-sampling the raw trajectory. By taking the resampled points as input, we apply PCA to standardize the geographic orientation, so that the intrinsic geometric shape is captured:  $\Delta p''_i = \text{PCA}(\Delta p'_i)$ . The output is the sequence of standardized points,  $[\Delta p''_1, \dots, \Delta p''_n]$ . Therefore, shape embedding becomes insensitive to length and direction, making it valuable for recognizing patterns regardless of the total distance traveled along the trajectory. Still, it remains sensitive to speed, thereby depicting the relationship between speed and the geographical shape of a trajectory.

**3.2 Complementarity by Detectors: Embedding Space vs. Input Space** To leverage the complementarity of the embedding and input spaces, we perform outlier detection in the embedding space to obtain anomaly scores ( $D_{emb}$ )<sup>1</sup>, as well as use the reconstruction loss of our autoencoder ( $D_{loss}$ ) which measures how well the model can reconstruct the original input.  $D_{emb}$  and  $D_{loss}$  are complementary, as anomalies can be divided into the following two scenarios:

(i) **Low score in  $D_{loss}$  but high score in  $D_{emb}$ .** When the embedding retains sufficient details, reconstruction loss is low but the anomaly is distinct in the embedding space. This case corresponds to Type B anomalies in Figure 3.

(ii) **High score in  $D_{loss}$  but low score in  $D_{emb}$ .** When the embedding fails to adequately represent the input, the reconstruction loss becomes an indicator of

<sup>1</sup>We employ LOF [5], although other effective tabular outlier detection algorithms can be used.

---

**Algorithm 1** Complementary Detector Elimination for Trajectory Anomaly Detection

---

```
1: Input:  $S = \{s_1, s_2, \dots, s_n\}$  (set of anomaly score lists)
2: Output:  $E$  (final set of selected score lists)
3: Initialize:  $E \leftarrow S$ 
4: while  $|S| > 1$  do
5:    $S \leftarrow S'$ 
6:   Compute  $\text{MAX}(S) := \max\{s_1(x), \dots, s_n(x)\}$  for all  $x$  (samples)
7:   for all  $s_i \in S$  do
8:      $\text{MAX}(S \setminus \{s_i\}) := \max\{s_j(x) \mid j \neq i\}$  for all  $x$ 
9:     Measure the ranking similarity between  $S$  and  $S \setminus \{s_i\}$ :
10:     $\text{JAC}_i := \text{Jaccard}(\text{MAX}(S), \text{MAX}(S \setminus \{s_i\}))$ 
11:   end for
12:   Identify the list  $s_i := \arg \min_i \text{JAC}_i$ 
13:    $S \leftarrow S \setminus \{s_i\}$ 
14:   Track  $\text{JAC}(t) := \text{Jaccard}(\text{MAX}(S), \text{MAX}(S \setminus \{s_i\}))$  for each iteration  $t$ 
15:   if  $\text{JAC}(t)$  shows a significant drop then
16:     Stop the elimination process
17:      $E \leftarrow S$ 
18:   end if
19: end while
20: Return  $E$ 
```

---

anomalies since they deviate from normal patterns and can not be regenerated accurately. Embeddings are less helpful because they might be too coarse to differentiate normal and anomalous data after projection. This case corresponds to the Type A anomalies in Figure 3.

In summary,  $D_{emb}$  and  $D_{loss}$  complement each other, which together ensure that we capture anomalies that either manifest clearly in the latent space or fail to be reconstructed accurately.

## 4 Ensembling Complementary Detectors

Given a dataset containing trajectory anomalies of certain type(s), our complementary base detectors may identify anomalies with varying accuracy. Thus, it is essential to effectively remove detectors that do not detect any for performance improvement. To this end, we propose an elimination-based method to retain complementary competent detectors by evaluating their influence and using changes in ranking similarity as the stopping criterion. The steps are outlined in Algorithm 1 and described in the following sections.

**4.1 Elimination Criterion** Our ensemble contains heterogeneous detectors (e.g., LOF scores on embeddings and reconstruction loss), where each detector has differ-

ent ranges of scores and interpretation. For unification, we first convert each detector’s scores to inverse ranks. Unlike previous consensus-based methods [15, 16], which iteratively add detectors to enhance agreement, we do not assume consensus among detectors. Poorly performing detectors are uncorrelated with well-performing ones, and even good detectors have only low correlation, as some anomalies may occur in multiple aspects. Hence, our approach adopts an elimination-base strategy [22].

Our elimination criterion is based on the impact of the removed detector on the ensemble’s rankings. At each iteration, we compute the Jaccard similarity between from all detectors and rankings excluding the  $i$ -th detector:

$$(4.1) \quad \text{JAC}_i = \text{Jaccard}(\text{MAX}(S), \text{MAX}(S \setminus \{s_i\})) .$$

Jaccard similarity emphasizes top-ranked anomalies, and MAX aggregation highlights anomalies even if detected by a single detector. We iteratively remove the detector with the lowest  $\text{JAC}_i$  as its removal causes the most significant impact on the ensemble rankings, indicating that it introduces noise rather than contributing valuable insight. Detectors with minimal impact (i.e., high Jaccard similarity) are likely redundant but not necessarily discarded.

**4.2 Stopping Criterion** As the elimination progresses, poorly performing detectors are removed. Eventually, two scenarios arise:

(i) Low Agreement: Complementary detectors have low agreement as they identify different anomalies. Removing a minority detector may resemble eliminating a "bad" detector due to its divergence.

(ii) High Agreement: Remaining detectors capture similar aspects, and further removals cause minimal ranking changes due to redundancy.

We stop the elimination when a significant drop in  $\text{Jaccard}(\text{MAX}(S), \text{MAX}(S \setminus \{s_i\}))$ , which indicates a complementary detector has been removed. Initially as poor detectors are removed, Jaccard similarity remains stable, but removing unique detectors causes a noticeable drop, reducing ensemble diversity and robustness. To automate this, we use the Kneedle algorithm [18] to detect the change point in the Jaccard similarity curve.

Once the elimination process is complete, we apply Average of Maximums (AOM) method over the remaining detectors. Consider the retained ensemble  $E$  consisting of detectors  $(s_1, s_2, \dots, s_m)$ ,  $s_i(x)$  represents the score given by the  $i$ -th detector for sample  $x$ . To obtain the final anomaly score for trajectory  $T$ , we divide the detectors into  $k$  subgroups, denoted as  $\{G_1, G_2, \dots, G_k\}$ , and compute the maximum score within each subgroup. The final ensemble score for  $T$  is then obtained by aver-

aging these maximum scores:

$$(4.2) \quad O_T = \frac{1}{k} \sum_{j=1}^k \max_{S_i \in G_j} S_i(T)$$

where  $\max_{S_i \in G_j} S_i(T)$  represents the maximum score for  $T$  within subgroup  $G_j$ . AOM balances performance and robustness [1], emphasizing the strongest anomaly signals while reducing noise from residual noisy detectors.

## 5 Experiments

### 5.1 Setup

Table 2: Summary statistics of datasets

Attribute	LA	Porto	Chengdu
Type	Synthetic	Real	Real
# of trajectories	859,907	302,822	229,007
Avg length	69.3	53.5	165.6
Sampling rate (s)	5	15	2~4

**Datasets** The experiments are conducted on three datasets, covering both synthetic and real-world trajectory data. **LA** consists of synthetic trajectories generated using the Data-Driven Trajectory Generator (DDTG) [3], a model-free and parameter-less method that uses aggregate origin-destination and traffic data to create realistic synthetic vehicle trajectories in Los Angeles. The second dataset **Porto** is a taxi trajectory dataset<sup>2</sup>, containing real-world taxi trajectories recorded from Porto, Portugal from 2013 to 2014. The third dataset is **Chengdu** which is a taxi trajectory dataset<sup>3</sup> provided by DiDi Chuxing, containing real taxi trajectories in Chengdu in August 2014. We use a 4-day subset from the one-month period. The statistics of these datasets can be found in Table 2.

**Settings** We preprocess the data by filtering out trajectories with fewer than 20 points. For LSTM training, the data is split 70/20/10 for training, validation, and testing, with a 2-layer architecture. For **Porto**, four LSTM models are trained per embedding type with sizes [4, 64, 128, 512], while **LA** and **Chengdu** use three models with sizes [16, 64, 128]. All models are trained on an NVIDIA RTX A6000 GPU.

**Groundtruth** We evaluate our method using both human-labeled and synthetic anomalies. For **Chengdu**, RL4OASD [24] manually labeled anomalies based on visual inspection, providing a real-world benchmark for naturally occurring anomalies. Additionally, we inject two types of synthetic anomalies, detour and route-switching, following previous works [12, 7]. Detour anomalies involve random lateral shifts of points in

Table 3: Anomaly detection results on Chengdu with human-labeled anomalies.

Method	AUROC	AUPR
IBAT	0.511	0.055
ATDRNN	0.502	0.050
GMVSAE	0.334	0.035
ATROM	0.600	0.071
RL4OASD	<b>0.835</b>	<b>0.154</b>
CETRAJAD	<u>0.675</u>	<u>0.124</u>

the trajectory while route-switching anomalies shift a continuous segment in a specific direction. Since prior works use grids to represent trajectories, we adapt their anomaly injection methods by applying shifts to grid sequences and using the center coordinates of each grid for CETRAJAD. During evaluation, we randomly sample 10,000 trajectories from test data and inject 5% anomalies. Further, we propose more diverse and realistic GPS-based anomaly designs.

**Baselines** Baselines include five trajectory anomaly detection methods: IBAT [23] detects anomalies using how much the target trajectory can be isolated from other trajectories. ATDRNN [19] uses RNN to obtain trajectory embeddings and learn to classify anomalies in a self-supervised manner. During training, it creates synthetic anomalies by adding noise to randomly sampled points in the trajectory. GMVSAE [12] uses a variational sequence autoencoder and a Gaussian mixture model to model the probability distribution of trajectories. After the VAE is trained, it detects anomalous trajectories by computing the likelihood of the target trajectory being generated from Gaussian components. ATROM [7] uses variational Bayesian methods to force the trajectories with distinct behaviors to follow different Gaussian priors. RL4OASD [24] matches trajectories to road networks and uses reinforcement learning to model the transition probability between road segments from historical trajectories. As the model outputs a binary label for each road segment, the anomaly score for the trajectory is computed as the mean of labels of all road segments in the trajectory. IBAT[23], ATDRNN[19], GMVSAE[12], and ATROM[7] convert raw trajectories into sequences of grid IDs, while RL4OASD[24] maps raw trajectories to road networks and use the sequences of edge IDs. We use a 100m by 100m grid for all datasets.

**Metrics** We employ two widely used metrics, Area Under the Receiver Operating Characteristic Curve (AUROC) and Area Under the Precision-Recall Curve (AUPR). AUROC measures the model’s ability to distinguish between normal and anomalous trajectories across different decision thresholds. AUPR is particularly suitable for imbalanced datasets, which is often the case in anomaly detection.

<sup>2</sup><https://www.kaggle.com/porto-taxi>

<sup>3</sup><https://outreach.didichuxing.com/research>

Table 4: Anomaly detection results for injected anomalies across Porto, LA, and Chengdu datasets.

Injection Type	Detour						Switch					
	Porto		LA		Chengdu		Porto		LA		Chengdu	
	AUROC	AUPR										
IBAT	0.823	0.182	<u>0.666</u>	<u>0.075</u>	0.823	0.182	0.709	<u>0.131</u>	<u>0.649</u>	<u>0.072</u>	0.770	0.158
ATDRNN	0.510	0.048	0.505	0.050	0.490	0.049	0.500	0.047	0.495	0.048	0.495	0.048
GMVSAE	0.589	0.063	0.584	0.069	0.599	0.065	0.604	0.068	0.547	0.060	0.560	0.057
RL4OASD	0.511	0.051	0.513	0.051	0.662	0.066	0.505	0.051	0.508	0.051	0.507	0.055
ATROM	<u>0.938</u>	<u>0.430</u>	0.620	0.068	<u>0.905</u>	<u>0.385</u>	<b>0.869</b>	<b>0.331</b>	0.550	0.053	<u>0.789</u>	<b>0.229</b>
CETRAJAD	<b>0.988</b>	<b>0.766</b>	<b>0.993</b>	<b>0.841</b>	<b>0.990</b>	<b>0.930</b>	<u>0.748</u>	0.113	<b>0.930</b>	<b>0.313</b>	<b>0.865</b>	<u>0.211</u>

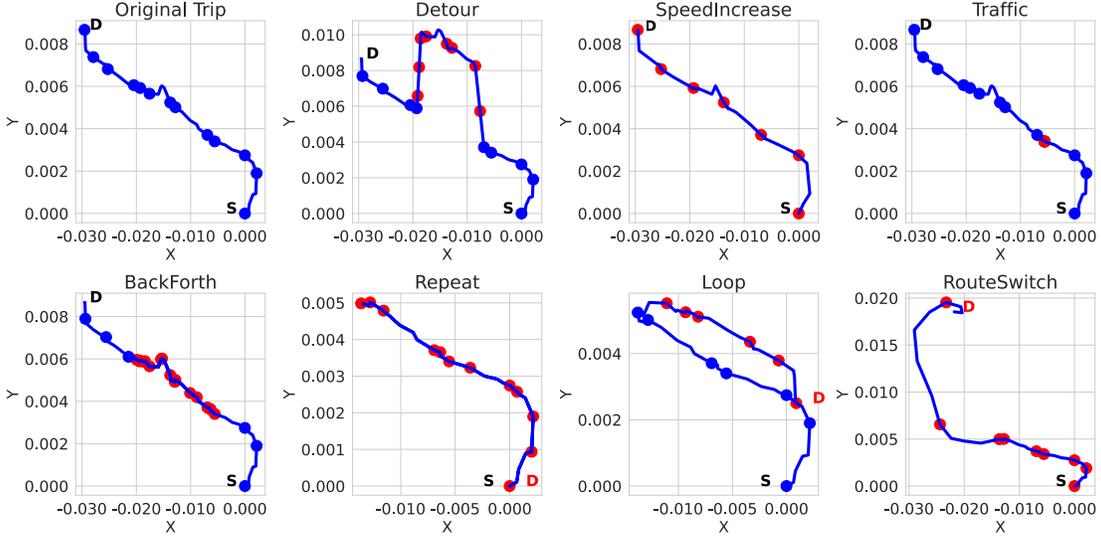


Figure 4: Visualization of synthetic anomalies. Blue lines show the trajectory. Markers are shown for every 5 timesteps. Red markers denote the injected anomalous points. S and D denote source and destination. If the destination is not the same as the original trajectory, we also show it in red.

**5.2 Results** Table 3 demonstrates that on the human-labeled real anomalies, CETRAJAD performs the second best compared to baselines. On Chengdu, RL4OASD [24] labels a trajectory as anomalous if it deviates significantly from the popular route between a given source and destination. As RL4OASD uses reinforcement learning to model the transition probabilities between locations within a trajectory, it is well-suited for detecting such deviations, which explains its best performance. In contrast, other baselines that rely on grid sequences, like GMVSAE and ATROM perform poorly, likely because the choice of grid size can lead to anomalous segments falling into the same grid, obscuring the anomalies.

Table 4 presents the results on synthetic anomalies. Results demonstrate that CETRAJAD performs well in most situations. It achieves the best performance (AUROC close to 1) for detour anomalies. Particularly for Chengdu, it achieves AUPR of 0.93, outperforming baselines significantly. One observation is that on LA, baseline methods do not perform well. This could be attributed to the larger geographical coverage of LA

compared to Porto and Chengdu, which results in a significantly larger number of grids. When using a grid size of 100m by 100m, the grid dimensions are (51, 118) for Porto, (83, 98) for Chengdu, but (693, 541) for LA. As many baseline methods rely on learning patterns from grid sequences, the increased sparsity of data in LA may lead to reduced performance. This again demonstrates that CETRAJAD performs well even when the geographic region is large.

### 5.3 Ablation Study

**5.3.1 Ensemble Methods** Previous studies primarily evaluate performance using a single type of anomaly (e.g., detour or route-switching). This limits the understanding of ensemble performance across a broader range of anomaly types. Hence we evaluate CETRAJAD with a more diverse set of anomalies in this section. Figure 4 shows our different designs of realistic trajectory anomalies. Since our injection methods use raw GPS coordinates while the trajectory anomaly detection

baselines use grids or road networks, a direct comparison would be unfair, as the choice of the grid size can significantly affect the perceived anomalousness of our injections. Therefore, in this section, we compare CETRAJAD only with simple ensemble methods (Mean and Max) and another selective consensus-based method (Mean-Ensemble). Mean-Ensemble sequentially removes the detector that agrees the least with the consensus at each iteration.

The results are shown in Table 5. With one injection type, we always inject SpeedIncrease anomalies, as they are primarily detected by Speed embeddings. Additional types are added in this order: Detour, BackForth, Traffic, RouteSwitch, Repeat, and Loop (visualization in Figure 4). We observe that as the number of anomaly types increases, all methods show declining performance due to the increased detection complexity. When only SpeedIncrease is injected, Mean-Ensemble outperforms CETRAJAD because Speed embeddings are effective in detecting the anomalies. Hence there is high consensus among detectors of Speed embeddings, which rank similar trajectories highly. However, as the number of types increases, CETRAJAD begins to outperform all three baselines. This indicates that CETRAJAD effectively leverages the complementarity among different detectors, which becomes crucial when the anomalies are more diverse.

**5.3.2 Complementary Components** We next examine the influence of the complementary components of CETRAJAD for Porto with 1, 3, 5, and 7 types of injected anomalies. We first evaluate the performance when using only  $D_{emb}$  or  $D_{loss}$ . Figure 5 presents the results. We observe that  $D_{emb}$  generally yields lower AUROC and AUPR compared to  $D_{loss}$ , particularly when the number of types is small. As the number of types increases, the performance of  $D_{emb}$  and  $D_{loss}$  becomes more similar. In all scenarios, using only  $D_{emb}$  or only  $D_{loss}$  does not outperform the combination of both. This indicates that while using only  $D_{loss}$  may be sufficient for detecting a single type of anomaly, employing both  $D_{loss}$  and  $D_{emb}$  provides better performance when dealing with a diverse set of anomalies.

The second study investigates the impact of Speed, Route, or Shape component. From Figure 6 when there is only one type of anomaly (SpeedIncrease), removing the Speed component (blue) results in a significantly lower AUPR compared to removing Route (orange) or Shape (green). As the number of types increases, removing any of the three components leads to similar performances. In all cases, using all three embeddings consistently performs better than using only two. The results demonstrate that Speed, Route, and Shape capture diverse information about trajectories by design.

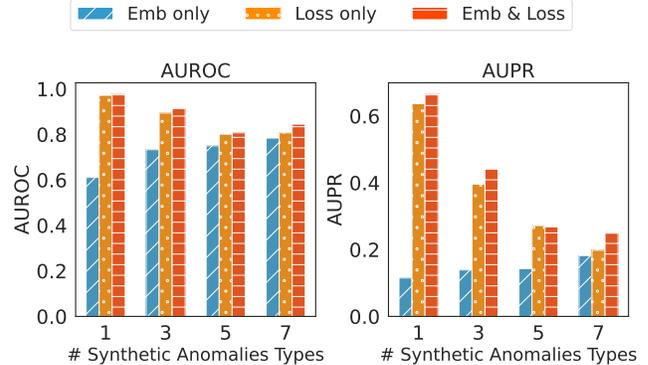


Figure 5: Performance of CETRAJAD when only  $D_{emb}$  or only  $D_{loss}$  is used.

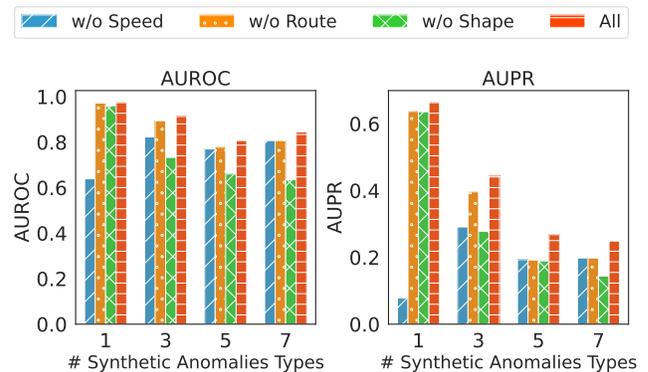


Figure 6: Performance of CETRAJAD when one of Speed, Route, and Shape is removed.

**5.3.3 Case Studies** In this section, we present case studies to demonstrate the effectiveness of the learned embeddings from detectors of different types.

In Figure 7, for the target trajectory, we display its nearest neighbors (NNs) based on different types of embeddings. From the Route visualization, we observe that the target trajectory and its NNs have similar lengths and directions as they all head to the northeast direction. In the Shape visualization, although the lengths and directions of the target trajectory and its NNs vary, their geometric shapes are similar: they all exhibit an inverted U-shape with a V-shaped tail at the end. The Speed visualization reveals that the target trajectory and its NNs have a similar number of timesteps (i.e., similar lengths) and share similar speed trends: slower at the beginning, faster in the middle, and slowing down towards the end. These visualizations demonstrate the effectiveness of the learned embeddings, as they successfully capture the characteristics they were designed for, as described in Table 1.

## 6 Related Work

**6.1 Trajectory Anomaly Detection** Existing studies on trajectory anomaly detection can be broadly cate-

Table 5: Anomaly detection results on Porto (left) and Chengdu (right) with injected anomalies under different ensemble methods.

Injection Type	Porto								Chengdu							
	1 Type		3 Types		5 Types		7 Types		1 Type		3 Types		5 Types		7 Types	
	AUROC	AUPR														
Mean	0.888	0.222	<u>0.889</u>	<u>0.291</u>	<u>0.806</u>	<u>0.208</u>	0.814	0.203	0.877	0.223	<b>0.893</b>	<b>0.289</b>	0.764	0.175	0.772	0.178
Max	0.868	0.191	0.854	0.230	0.774	0.172	0.784	0.168	0.823	0.179	0.846	0.226	0.733	0.141	0.747	0.153
Mean-Ensemble	<b>0.980</b>	<b>0.742</b>	0.829	0.274	0.783	0.199	<u>0.822</u>	<u>0.212</u>	<u>0.926</u>	<b>0.348</b>	0.830	0.218	<b>0.779</b>	<b>0.186</b>	<b>0.777</b>	<b>0.183</b>
CETRAJAD	<u>0.976</u>	<u>0.667</u>	<b>0.913</b>	<b>0.443</b>	<b>0.807</b>	<b>0.268</b>	<b>0.846</b>	<b>0.249</b>	<b>0.928</b>	<u>0.291</u>	<u>0.852</u>	<b>0.291</b>	<u>0.759</u>	<b>0.221</b>	<u>0.773</u>	<b>0.193</b>

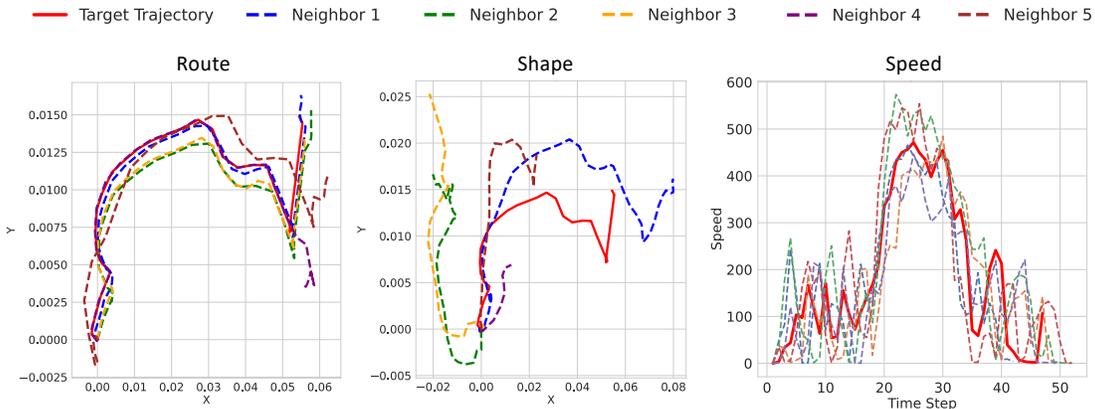


Figure 7: Visualization of nearest neighbors (NNs) of the target trajectory from different types of embeddings. Route and Shape embeddings display the trajectories (in normalized coordinates), while Speed visualization shows the speed over time (x-axis: timesteps, y-axis: speed per timestep). The target trajectory is always shown in red while its NNs are shown in different colors.

gorized into distance-based and learning-based methods. Distance-based methods measure the deviation of a trajectory from predefined "normal" trajectories, often the most frequent routes, relying on the distance, pairwise comparisons or trajectory isolation [11, 23, 6]. These methods, while effective, are computationally expensive. Learning-based methods use machine learning, particularly sequence modeling, to model trajectory patterns. Early works [19] applied supervised RNNs, while more recent approaches [12, 7] leverage variational autoencoders to identify anomalies as deviations from Gaussian priors. Reinforcement learning has also been employed for sub trajectory detection [24]. Most methods convert raw GPS coordinates into grid cells or road networks, which can limit performance in areas without well-defined roads (e.g., animal migration). Additionally, grid size affects learning, with smaller grids causing sparsity and larger grids losing detail.

**6.2 Outlier Score Ensemble** Outlier detection methods often suffer from biases in individual detectors, leading to over- or underestimation of anomalies. Ensemble approaches aggregate detector outputs to improve robustness. Simple aggregation methods (e.g., averaging, maximization, and Average-of-Maximum) enhance

robustness but fail to address weak detectors effectively [2, 1]. Selective models like SELECT [15] and CARE [16] construct pseudo ground truth or use weighted correlations for detector selection. LSCP [26] further enhances selection using data locality. However, these methods often rely on consensus-based selection, overlooking the complementarity between detectors. Our approach addresses this gap by focusing on complementarity to improve robustness and accuracy in anomaly detection.

## 7 Conclusion

We introduce CETRAJAD, which detects trajectory anomalies across multiple aspects and offers a novel perspective on ensemble methods by emphasizing complementarity. Experimental results demonstrate that CETRAJAD performs effectively on both labeled real-world anomalies and injected synthetic anomalies. Additionally, our GPS-based anomaly injection method provides a versatile solution for evaluating anomaly detection models, addressing challenges posed by existing grid- or road network-based approaches. By relying solely on raw GPS coordinates, CETRAJAD is flexible and generalizable to other trajectory types, including animal migration, airplanes, and marine vessels.

## Acknowledgments

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0033. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government. The authors thank Lingxiao Zhao for their suggestions on data representation for modeling and Haomin Wen for their feedback on the manuscript.

## References

- [1] C. C. AGGARWAL, *Outlier Ensembles*, Springer International Publishing, Cham, 2017, pp. 185–218.
- [2] C. C. AGGARWAL AND S. SATHE, *Theoretical foundations and algorithms for outlier ensembles*, SIGKDD Explor. Newsl., 17 (2015), p. 24–47.
- [3] C. ANASTASIOU, S. H. KIM, AND C. SHAHABI, *Generation of synthetic urban vehicle trajectories*, in 2022 IEEE International Conference on Big Data (Big Data), 2022, pp. 359–366.
- [4] A. BELHADI, Y. DJENOURI, J. C.-W. LIN, AND A. CANO, *Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges*, ACM Transactions on Management Information Systems (TMIS), 11 (2020), pp. 1–29.
- [5] M. M. BREUNIG, H.-P. KRIEGEL, R. T. NG, AND J. SANDER, *Lof: identifying density-based local outliers*, SIGMOD '00, New York, NY, USA, 2000, Association for Computing Machinery, p. 93–104.
- [6] C. CHEN, D. ZHANG, P. S. CASTRO, N. LI, L. SUN, S. LI, AND Z. WANG, *iboot: Isolation-based online anomalous trajectory detection*, Trans. Intell. Transport. Syst., 14 (2013), p. 806–818.
- [7] Q. GAO, X. WANG, C. LIU, G. TRAJCEVSKI, L. HUANG, AND F. ZHOU, *Open anomalous trajectory recognition via probabilistic metric learning*, in Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23, 2023.
- [8] X. HAN, R. CHENG, C. MA, AND T. GRUBENMANN, *Deeptea: effective and efficient online time-dependent trajectory outlier detection*, Proc. VLDB Endow., 15 (2022), p. 1493–1505.
- [9] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural Comput., 9 (1997), p. 1735–1780.
- [10] R. JIAO, J. BAI, X. LIU, T. SATO, X. YUAN, Q. A. CHEN, AND Q. ZHU, *Learning representation for anomaly detection of vehicle trajectories*, in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 9699–9706.
- [11] J.-G. LEE, J. HAN, AND X. LI, *Trajectory outlier detection: A partition-and-detect framework*, in 2008 IEEE 24th International Conference on Data Engineering, 2008, pp. 140–149.
- [12] Y. LIU, K. ZHAO, G. CONG, AND Z. BAO, *Online anomalous trajectory detection with deep generative sequence modeling*, in 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020, pp. 949–960.
- [13] M. MEMARZADEH, B. MATTHEWS, AND T. TEMPLIN, *Multi-Class Anomaly Detection in Flight Data Using Semi-Supervised Explainable Deep Learning Model*, 2021.
- [14] F. MENG, G. YUAN, S. LV, Z. WANG, AND S. XIA, *An overview on trajectory outlier detection*, Artificial Intelligence Review, 52 (2019), pp. 2437–2456.
- [15] S. RAYANA AND L. AKOGLU, *Less is more: Building selective anomaly ensembles*, ACM Trans. Knowl. Discov. Data, 10 (2016).
- [16] S. RAYANA, W. ZHONG, AND L. AKOGLU, *Sequential ensemble learning for outlier detection: A bias-variance perspective*, in 2016 IEEE 16th International Conference on Data Mining (ICDM), 2016, pp. 1167–1172.
- [17] A. SALAZAR MIRANDA, Z. FAN, F. DUARTE, AND C. RATTI, *Desirable streets: Using deviations in pedestrian trajectories to measure the value of the built environment*, Computers, Environment and Urban Systems, 86 (2021), p. 101563.
- [18] V. SATOPAA, J. ALBRECHT, D. IRWIN, AND B. RAGHAVAN, *Finding a "kneedle" in a haystack: Detecting knee points in system behavior*, in 2011 31st International Conference on Distributed Computing Systems Workshops, 2011, pp. 166–171.
- [19] L. SONG, R. WANG, D. XIAO, X. HAN, Y. CAI, AND C. SHI, *Anomalous trajectory detection using recurrent neural network*, in Advanced Data Mining and Applications: 14th International Conference, ADMA 2018, Nanjing, China, November 16–18, 2018, Proceedings, Berlin, Heidelberg, 2018, Springer-Verlag, p. 263–277.
- [20] Y. SU, D. YAO, X. ZHOU, Y. ZHANG, Y. FAN, L. BAI, AND J. BI, *Tripsafe: Retrieving safety-related abnormal trips in real-time with trajectory data*, in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 2446–2450.
- [21] J. WIEDERER, A. BOUAZIZI, M. TROINA, U. KRESSEL, AND V. BELAGIANNIS, *Anomaly detection in multi-agent trajectories for automated driving*, in 5th Annual Conference on Robot Learning, 2021.
- [22] J. YANG, S. RAHARDJA, AND S. RAHARDJA, *Foor: Be careful for outlier-score outliers when using unsupervised outlier ensembles*, IEEE Transactions on Computational Social Systems, 11 (2024), pp. 2843–2852.
- [23] D. ZHANG, N. LI, Z.-H. ZHOU, C. CHEN, L. SUN, AND S. LI, *ibat: detecting anomalous taxi trajectories from gps traces*, in Proceedings of the 13th International Conference on Ubiquitous Computing, UbiComp '11,

New York, NY, USA, 2011, Association for Computing Machinery, p. 99–108.

- [24] Q. ZHANG, Z. WANG, C. LONG, C. HUANG, S. YIU, Y. LIU, G. CONG, AND J. SHI, *Online anomalous subtrajectory detection on road networks with deep reinforcement learning*, in 2023 IEEE 39th International Conference on Data Engineering (ICDE), Los Alamitos, CA, USA, apr 2023, IEEE Computer Society, pp. 246–258.
- [25] Z. ZHANG, G. QI, A. A. CEDER, W. GUAN, R. GUO, AND Z. WEI, *Grid-based anomaly detection of freight vehicle trajectory considering local temporal window*, Journal of Advanced Transportation, 2021 (2021), p. 8103333.
- [26] Y. ZHAO, Z. NASRULLAH, M. K. HRYNIEWICKI, AND Z. LI, *LSCP: Locally Selective Combination in Parallel Outlier Ensembles*, Society for Industrial and Applied Mathematics, May 2019, p. 585–593.