International Journal of Data Warehousing and Mining
Volume 19 • Issue 2

RETAD:
Vehicle Trajectory Anomaly Detection Based on Reconstruction Error

Chaoneng Li, Xidian University, China
https://orcid.org/0000-0002-2781-4683

Guanwen Feng, Xidian University, China

Yiran Jia, Xidian University, China

Yunan Li, Xidian University, China

Jian Ji, Xidian University, China

Qiguang Miao, Xidian University, China*

ABSTRACT

Due to the rapid advancement of wireless sensor and location technologies, a large amount of mobile agent trajectory data has become available. Intelligent city systems and video surveillance all benefit from trajectory anomaly detection. The authors propose an unsupervised reconstruction error-based trajectory anomaly detection (RETAD) method for vehicles to address the issues of conventional anomaly detection, which include difficulty extracting features, are susceptible to overfitting, and have a poor anomaly detection effect. RETAD reconstructs the original vehicle trajectories through an autoencoder based on recurrent neural networks. The model obtains moving patterns of normal trajectories by eliminating the gap between the reconstruction results and the initial inputs. Anomalous trajectories are defined as those with a reconstruction error larger than anomaly threshold. Experimental results demonstrate that the effectiveness of RETAD in detecting anomalies is superior to traditional distance-based, density-based, and machine learning classification algorithms on multiple metrics.

KEYWORDS

Anomaly Detection, Autoencoder, Reconstruction Error, Recurrent Neural Networks, Vehicle Trajectory

INTRODUCTION

With the continuous progress of Global Positioning Systems (GPS), laser radar, high-resolution cameras, and other sensor technologies, the trajectory information of a huge number of mobile individuals has grown drastically, including pedestrian trajectories, vehicle trajectories, ship and aircraft trajectories,

DOI: 10.4018/IJDWM.316460
*Corresponding Author

and so on. Anomaly detection based on the trajectory of the above mobile agents is widely applied in a variety of visual tasks. For instance, in the field of intelligent traffic monitoring, anomalous vehicle behavior, such as sudden acceleration, deviation from the road, position drift, etc., may present dangers related to drunk driving, traffic collisions, and road violations. Timely identification and assessment of anomalous activity within the region of interest is crucial for proactive resolution measures and represents a higher level of object perception (Tu et al. 2017). Due to the overload of trajectory information and differences in camera positions, sampling frequencies, and scene structures in real traffic monitoring scenarios, vehicle trajectories usually differ in time and space characteristics, and anomaly detection tasks cannot be accomplished by manual operations. Consequently, it is significant to design an automatic anomaly detection algorithm to mine and analyze traffic trajectories.

The objective of vehicle trajectory anomaly detection is to identify abnormal trajectories from a large amount of unlabeled traffic trajectory data from the real world or virtual simulation. Chandola et al. (2009) developed the definition of anomalies, which are irregularities in data sets that do not adhere to acceptable behavior. Specific to the field of traffic, anomalous trajectories generally include drastic vehicle speed changes caused by complex and changeable traffic conditions; the vehicle trajectory deviates from the road line for a long time, or the vehicle position drifts caused by the driver's improper operation; the vehicle violates the traffic rules and drives in the opposite direction.

BACKGROUND AND RELATED WORK

In general, trajectory anomaly detection studies mainly include detection methods based on classification and clustering (Piciarelli et al. 2008, Yang et al. 2013, Li et al. 2007, Zhu et al. 2017, Kumar et al. 2017, Lv et al. 2017), methods based on distance and density (Lee et al. 2008, Liu et al. 2012b, 2013, San Román et al. 2019, Luan et al. 2017, Huang and Zhang 2019, Tang and Ngan 2016), and methods based on machine learning and pattern learning (Song et al. 2018, Ma et al. 2018, Bouritsas et al. 2019, Liu et al. 2020, Fu et al. 2020, Liatsikou et al. 2021).

The classification-based anomaly detection method first trains the classification model and then uses the pretrained model to judge whether the trajectory to be evaluated is anomalous. Piciarelli et al. (2008) suggested an anomalous trajectory detection method based on a support vector machine (SVM). In this method, every trajectory was described by a defined dimensional characteristic vector of the consistent samples of the original trajectory, and the trajectory classification was completed without prior information on the distribution of trajectory anomalies. However, classification-based methods need to annotate data for model training, which leads to a large amount of labor and time expenditure and reduces the usability of such methods to some extent. The clustering-based anomaly detection method clusters all trajectories into multiple groups. Dense classes are regarded as normal trajectories, and sparse classes are regarded as anomalous trajectories. Kumar et al. (2017) presented a two-phase clustering algorithm, in which trajectories were grouped by similarity measure while considering the trajectory direction information, and then the clustering results with fewer trajectories were judged as anomalous.

The primary premise of distance-based trajectory anomaly detection approaches is that if a trajectory is far away from most other trajectories, it may be an anomalous trajectory. San Román et al. (2019) proposed a novel context-aware distance (CaD), which was constructed from the weighted average of trajectory angle differences, the Euclidean distance, and the number of points on trajectories. Based on the CaD distance, an unsupervised technique was proposed to identify anomalous pedestrian trajectories extracted from a video surveillance system. Density-based approaches are tightly linked to distance-based trajectory anomaly detection methods because density is frequently described by various distances. The key concept behind density-based methods is that trajectories in low-density regions are judged as anomalies, while normal trajectories appear in relatively dense regions. Luan et al. (2017) presented a local density-based trajectory anomaly detection method called LDTRAOD. The algorithm used a segmentation and detection framework to calculate the local density of each divided trajectory and then calculated its local anomaly factor according to the local density. If the local anomaly factor was greater than the detection threshold, the divided trajectory was marked as abnormal. However, distance- and density-based methods must continuously query the entire historical trajectory database to compare metrics, and the computational cost is relatively high.

With the advancement of deep learning technology, neural networks have been proven to be able to automatically learn temporal and spatial features from massive time sequence data; therefore, they are also commonly utilized in trajectory anomaly detection. Methods based on trajectory pattern learning usually rely on recurrent neural networks (RNNs), such as long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) or gated recurrent unit (GRU) (Cho et al. 2014), to complete trajectory temporal modeling and feature learning. By training a reconstruction model, such as an autoencoder (AE) (Hinton and Zemel 1993) or generative adversarial network (GAN) (Goodfellow et al. 2014), the reconstructed trajectory is compared with the original input trajectory to detect anomalous segments. Song et al. (2018) suggested an RNN-based anomaly detection method, ATD-RNN. ATD-RNN used RNN to obtain the sequence information and describe the intrinsic features between anomalous and normal trajectories and alleviated the potential data sparsity problem by expanding many historical trajectories.

As shown in previous work, an increasing number of anomaly detection algorithms have been presented in trajectory mining, but the detection effect of these methods largely depends on data sets and parameter settings. In connection with the real world, it can be found that the significant barrier of anomaly detection mainly occurs in data collection. Due to the low frequency of anomalous events, the proportion of positive samples and negative samples in the training data is highly uneven. Currently, the available open source data sets for trajectory anomaly detection are limited. Accessible data sets, such as TRAFFIC (Lin et al. 2016) and CROSS (Morris and Trivedi 2011) are only collected and annotated in specific traffic environments. In addition, the trajectory anomaly detection algorithm needs to set a reasonable anomaly threshold and detection subsequence length to achieve optimal performance. Based on the above analysis, researchers prefer to adopt unsupervised algorithms for modeling anomaly detection problems. First, the model is learned from the normal trajectories. Then, the labeled validation set is employed to determine the anomaly threshold and other hyperparameters. Finally, the trained model is evaluated on the independent testing set.

Inspired by the application of deep neural networks, we provide a reconstruction error-based trajectory anomaly detection (RETAD) method for vehicles. RETAD takes the RNN-based autoencoder as the main architecture. Specifically, the GRU is used as the fundamental unit of the encoder and decoder of the autoencoder. The model is trained unsupervised, the output of the model is the reconstruction result of the input trajectory data, and trajectories with reconstruction errors larger than the anomaly threshold are judged to be anomalous. At the same time, a global pooling mechanism is added in the encoding and decoding process to model the interaction between vehicle trajectories to improve the effect of trajectory reconstruction. In addition, for the trajectory data in the algorithm validation and testing stages, compared with manually labeling anomalous trajectories, we employ the data-driven simulation algorithm proposed in Chao et al. (2017) to generate trajectories with various kinds of anomalies by controlling the parameter configuration of the algorithm.

We summarize our main contributions as follows:

- An unsupervised learning method is introduced in vehicle trajectory anomaly detection, and the model training is completed under real-world traffic trajectory data, which avoids the overfitting problem caused by unbalanced training trajectories.
- Combining the restoration ability of the autoencoder and the time series modeling advantage of the RNN, through the global pooling mechanism, the interaction between trajectories is fully considered, and an anomaly detection model RETAD based on trajectory reconstruction errors is provided.
- The test results on various vehicle trajectory data containing different types of anomalies reveal that our proposed RETAD model outperforms both traditional distance- and density-based methods and machine learning methods in terms of anomaly detection evaluation metrics.

Trajectory-Related Definitions

Traffic trajectory data record the spatial position information of vehicles at different times, there is a close context relationship between adjacent points, and the trajectory is a collection of these ordered points.

Definition 1 (Trajectory Point): Traffic trajectory points are GPS coordinates collected by navigation apps (e.g., Google Maps and Baidu Maps), ride-sharing platforms (e.g., Uber and Didi Chuxing), and GPS-equipped vehicles (e.g., Apollo and Waymo). A spatiotemporal trajectory point can be represented as $p = \left(x, y, t\right)$, including the position information $\left(x, y\right)$ and the timestamp t.

Definition 2 (Trajectory): The trajectory is a sequence composed of spatiotemporal trajectory points and can be expressed as $T = \left\langle p_1, p_2, \ldots, p_k \right\rangle$, where $p_i \left(i \in \left\{1, 2, \ldots, k\right\}\right)$ represents a trajectory point and k denotes the total number of trajectory points.

Definition 3 (Trajectory Data sets): The trajectory data set is a collection of trajectories, expressed as $T = \left\{T_1, T_2, \ldots, T_n\right\}$, where n is the number of trajectories.

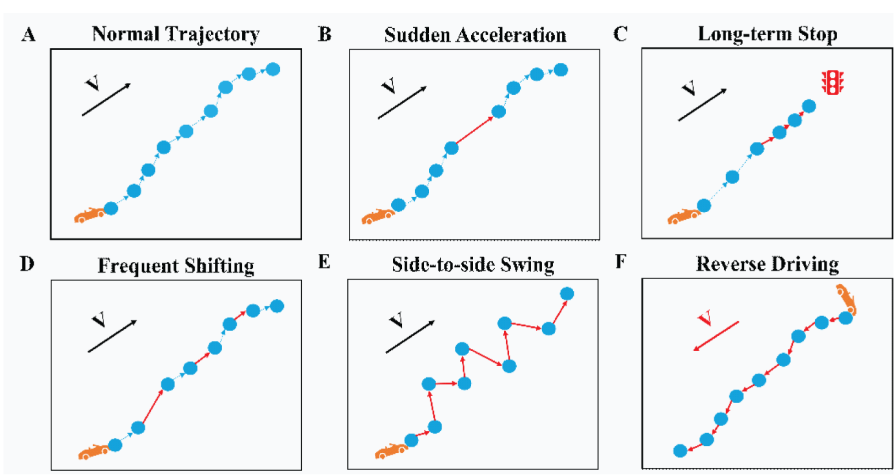
Definition 4 (Anomalous Trajectory): According to the traffic rules, driving habits in the real world, and changes in the vehicle driving speed and direction, we define the anomalous vehicle trajectories into five categories: sudden acceleration, long-term stop, frequent shifting, side-to-side swing, and reverse driving, as shown in Figure 1.

MATERIALS AND METHODS

Data Sets

In this paper, a training data set containing 5,512 vehicles and a total of approximately 22,000 trajectories is constructed for training the anomaly detection model RETAD, which is extracted from the real-world traffic flow data set Next Generation Simulation (NGSIM) (Thompson 2016). NGSIM has been widely used in the field of traffic simulation. It consists of the vehicle dynamic information collected by cameras along the road in different periods, including vehicle speed, acceleration, position coordinates, etc. To validate the anomaly detection performance of RETAD, we employ

Figure 1. Typical Categories of Anomalous Trajectories



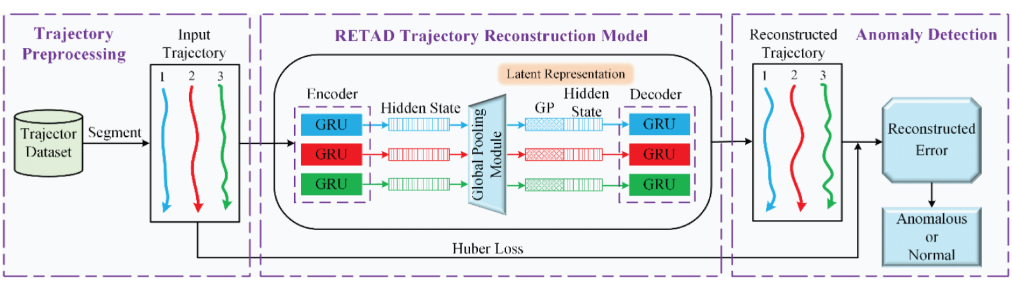
Note: The red mark indicates abnormal behavior. The arrows with "v" point to the directions of vehicles.

the data-driven texture synthesis method proposed in Chao et al. (2017) to synthesize six segment trajectories containing different types of anomalies (see Figure 1) as testing data sets, named Traj-1, Traj-2..., Traj-6. The virtual traffic simulation method proposed in Chao et al. (2017) can preserve the temporal and spatial characteristics of the input traffic flows. The interaction between vehicles in the scene strictly follows the traffic rules and can remain synchronized with the adjacent vehicles, which ensures the authenticity of the synthetic data. The number of trajectories in the synthesized testing data sets ranged from 1,000 to 2,500, in which the proportion of anomalous trajectories was approximately 5%, and the number of anomalous trajectories steadily increased from Traj-1 to Traj-6. Furthermore, the trajectories involved in the experiments are all first-order differences of local coordinates, and the frequency of trajectory acquisition is unified to 10 frames per second (10 fps).

METHODOLOGY

In this paper, we design an unsupervised anomaly detection model RETAD based on trajectory reconstruction error with the help of an autoencoder and recurrent neural network, in which the encoder and decoder modules of the autoencoder are implemented by GRU. The structure of RETAD and anomaly detection pipeline are shown in Figure 2. Our anomaly detection process includes three steps. First, in the trajectory preprocessing stage, the trajectories to be measured in the database are divided into fixed-length subtrajectories, and the position information of trajectory points $\left(x, y\right)$ is used as the input of the next step. Then, the RETAD model is used to encode the input trajectories and extract features. Next, a global pooling module (GPM) is introduced to model the interaction of all vehicles in the current scene, and the hidden state and the output of the GPM (GP) are concatenated to obtain the final latent representation as the input of the decoder to complete the trajectory reconstruction. Finally, in the anomaly detection stage, the reconstruction error of the trajectory is taken as the basis for anomaly judgment, and the trajectories with reconstruction errors larger than the anomaly threshold are judged as anomalous. Otherwise, it is classified as a normal trajectory. The RNN-based encoder and decoder in the RETAD fully consider the motion characteristics of vehicle trajectories in the observed time period and can learn the morphological differences between normal and anomalous trajectories. The model adopts an unsupervised training manner, and the training set may also contain a small number of anomalous trajectories, which can avoid the overfitting problem caused by unbalanced training samples.

Figure 2. Pipeline of the RETAD Algorithm



Note: RETAD includes three steps: trajectory preprocessing, trajectory reconstruction, and anomaly detection. Global Pooling(GP), Gated Recurrent Unit (GRU), Reconstruction Error-based Trajectory Anomaly Detection (RETAD).

Autoencoder

The objective of the autoencoder is to learn the latent representation of trajectories in an unsupervised manner. The autoencoder consists of an encoder and a decoder network. The encoder receives the original trajectory input and extracts the fixed-size features as a latent representation, which is further used as the input of the decoder network. The decoder network reconstructs the original input trajectory through training. For convenience, let $x = \left\{x^1, x^2, x^3, \ldots, x^t, \ldots, x^N\right\}$ denote the input trajectory and $\hat{x} = \left\{\hat{x}^1, \hat{x}^2, \hat{x}^3, \ldots, \hat{x}^t, \ldots, \hat{x}^N\right\}$ denote the reconstructed trajectory. Here, $x^t \in \mathbb{R}^2$ is the two-dimensional (2D) position at the t-th timestamp, and N denotes the length of the trajectory. The encoding phase is responsible for extracting features from the input trajectory, and a hidden state h_E^t at timestamp t is dependent on the input position information x^t and the hidden state at the previous timestamp h_E^{t-1}. The latent representation h_E of x is fed into the decoder to complete reconstruction. In the decoding phase, each hidden state h_D^t produces a reconstructed position $\hat{x}^t$ depending on the previous decoder state h_D^{t-1} as well as the previous reconstructed output $\hat{x}^{t-1}$. The encoding and decoding process of the AE can be expressed as:

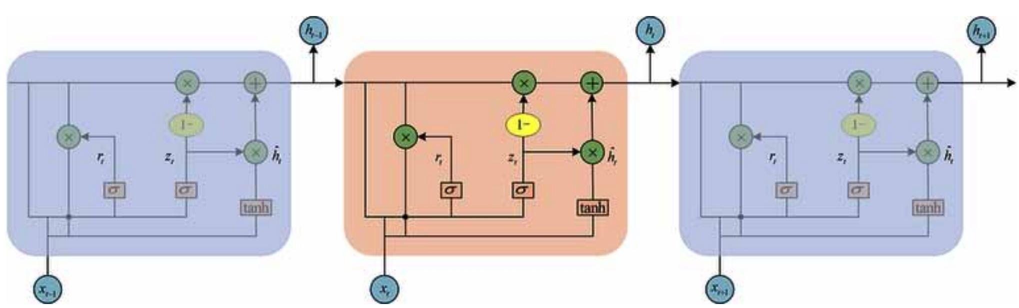
$$
\begin{aligned}
h_E &= \sigma_1\left(w_1^T x + b_1\right) \\
\hat{x} &= \sigma_2\left(w_2^T h_E + b_2\right)
\end{aligned}
$$

where h_E is the output of the encoder, σ_1 and σ_2 are the activation functions, w_1^T and w_2^T are the weights of the encoder and decoder, and b_1 and b_2 are bias vectors. The purpose of autoencoder training is to minimize the difference between $\hat{x}$ and x.

GRU

RNN has become the most popular neural network architecture for handling sequence analysis. The typical recurrent neural network, on the other hand, has limits for long-term prediction due to faults such as gradient disappearance and explosion. The LSTM and GRU models are RNN variations that have been shown to overcome the concerns listed above. The LSTM and GRU share many of the same underlying ideas. To preserve as much long-term knowledge as feasible, they all employ gated techniques. The LSTM model, on the other hand, takes more time to train due to its complicated structure, whereas the GRU model has a simpler structure and fewer parameters and can be trained quickly. To derive temporal relationships from trajectories, we adopt the GRU model. The GRU network structure is illustrated in Figure 3.

Figure 3. The Architecture of the Gated Recurrent Unit



The GRU model has two gates: a reset gate and an update gate. The reset gate r_t determines how the incoming data information is combined with the previous memory. The update gate z_t controls the amount of information that the prior memory stores to the current time step. The update process of the GRU may be expressed as follows:

$$\begin{aligned} z_t &= \sigma\left(w_z \cdot \left[h_{t-1}, x_t\right]\right) \\ r_t &= \sigma\left(w_r \cdot \left[h_{t-1}, x_t\right]\right), \\ \hat{h}_t &= \tanh\left(w_h \cdot \left[r_t * h_{t-1}, x_t\right]\right) \\ h_t &= \left(1 - z_t\right) * h_{t-1} + z_t * \hat{h}_t \end{aligned}$$

where x_t is the input vehicle position at the current timestep, h_t is the output of the GRU, σ is the activation function, w_z, w_r, and w_h are the related weights, and $\hat{h}_t$ denotes the candidate hidden state.

Global Pooling Module

We introduce a global pooling module, as shown in Figure 4, between the encoder and the decoder (see Figure 2) to realize information sharing between the GRUs to model the vehicle interaction.

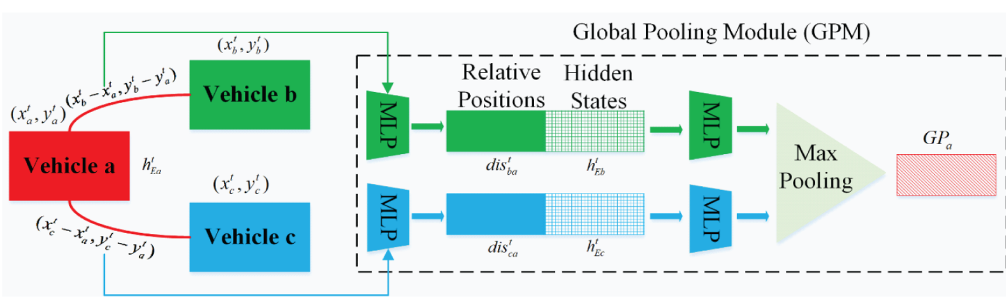
Taking vehicle a in Figure 4 as an example, the trajectory reconstruction of the RETAD model can be described as follows:

$$h_{Ea}^t = GRU\left(h_{Ea}^{t-1}, x_a^t; w_{encoder}\right)$$

where x_a^t is the position of vehicle a at timestamp t, which is processed by an embedding function. The GRU weights $w_{encoder}$ of the encoder are shared between all vehicles. Similarly, the hidden states h_{Eb}^t and h_{Ec}^t of vehicles b and c can be obtained:

$$GP_a^t = GPM\left(h_{Ea}^t, \left[dis_{ba}^t; h_{Eb}^t\right], \left[dis_{ca}^t; h_{Ec}^t\right]\right)$$

Figure 4. Global Pooling Module



Note: The relative positions between vehicle a and all other vehicles in the current scene are computed. Then, according to this relative position information and the respective hidden states, the pooling vector GP_a of vehicle a is computed by max pooling. Multilayer Perceptron (MLP).

where the global pooling module is implemented by max pooling, dis_{ba}^{t} and dis_{ca}^{t} are the relative position information of vehicles b and c with respect to a, which are processed by MLP, and [;] denotes concatenation:

$$h_{Da}^{t} = GRU\left(MLP\left(GP_{a}^{t}, h_{Da}^{t-1}\right), x_{a}^{t}; w_{decoder}\right)$$
$$\hat{x}_{a}^{t} = MLP\left(h_{Da}^{t}\right)$$

where $w_{decoder}$ are the GRU weights of the decoder and $\hat{x}_{a}^{t}$ is the reconstructed trajectory.

Anomaly Detection

The goal of RETAD model training is to minimize the divergence between the reconstruction result and the initial input as much as possible. The optimization process may be described as follows:

$$w_*, b_* = \underset{w_*, b_*}{\arg\min} \parallel x - \hat{x} \parallel$$

To accomplish the minimization task above, we use Huber loss (Huber 1992) to calculate the reconstruction error between $\hat{x}$ and x. Huber loss can enhance the robustness of the mean square error (MSE) loss to noise and reduce the degree of penalty for anomalous points. The loss function of the RETAD model is defined as:

$$Loss_{\delta}\left(x, \hat{x}\right) = \begin{cases} \frac{1}{2}(x - \hat{x})^2, \left|x - \hat{x}\right| \le \delta \\ \delta \cdot \left|x - \hat{x}\right| - \frac{1}{2}\delta^2, \left|x - \hat{x}\right| > \delta \end{cases}$$

where δ is a hyperparameter, which is specified as 1.0 in our experiment. Since the trajectory needs to be divided by a fixed length before reconstruction, the average reconstruction error (ARE) of the entire traffic flow is calculated to comprehensively measure the reconstruction effect:

$$ARE = \frac{1}{M}\sum_{i=1}^{M} Loss_{\delta}^{i}\left(x, \hat{x}\right)$$

where $Loss_{\delta}^{i}\left(x, \hat{x}\right)$ is the Huber loss of the i-th subtrajectory and M represents the total number of subtrajectories.

Evaluation Metrics

In essence, detecting trajectory anomalies is a binary classification task. Therefore, according to the ground truth and model prediction, there are four detection outcomes of the algorithm: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The specific classification results are shown in Table 1.

In this research, we assess the performance of multiple anomaly detection techniques using basic machine learning measurement methods, such as accuracy (ACC), precision (P), recall (R), and F1-score (F1). The percentage of properly identified trajectories to all trajectories is expressed by ACC. R

Table 1. Confusion Matrix of Trajectory Anomaly Detection

Ground truth	Prediction results	
	Anomalous trajectory	Normal trajectory
Anomalous trajectory	TP	FN
Normal trajectory	FP	TN

is the ratio of accurately recognized anomalous trajectories to all actual anomalous trajectories, and P is the proportion of successfully classified anomalous trajectories to the overall forecasted anomalous trajectories. In the anomalous trajectory detection task, the proportion of normal trajectories to the total data is far greater than that of anomalous trajectories, and the goal of our method is to identify as many anomalous trajectories as possible. Consequently, our focus is on the metric R. In addition, F1 is a more appropriate measurement than ACC for unbalanced binary data sets since it considers both accuracy and recall. The calculations of the above metrics are as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

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

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

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

RESULTS

Experiment Settings

We adopted PyTorch (Paszke et al. 2019) to build the RETAD model, and we trained the model on an NVIDIA GeForce RTX 3090. The following parameters are used to improve our model: the hidden state dimensions for the encoder and decoder are 32 and 64, the number of GRU layers is set to 2, the dropout ratio equals 0.5, and RMSProp (Ruder 2016) is employed to optimize RETAD with a learning rate of 0.0001. Two other important parameters are anomaly threshold $\mu = 0.4$ and subtrajectory length $l = 20$.

Baselines

We select several traditional typical methods based on distance and density, machine learning, and recent pattern learning as baselines and compare them with the RETAD model:

1. **Longest Common Subsequence (LCSS) (Vlachos et al. 2002):** LCSS detects anomalies based on the similarity measured by LCSS distances between testing trajectory and overall trajectories in the training data set. Therefore, this method needs to continuously query and compare the training set, resulting in high computational resource overhead.
2. **TOP-EYE (Ge et al. 2010):** TOP-EYE considers anomalous trajectories in direction and density. The algorithm discretizes the successive region into a microscopic grid, and the count of trajectories passing through the grid is used to determine the density of trajectories inside every grid. As a result, the anomalous grade of an incoming trajectory may be calculated based on the density of the trajectory in the grid that it truly goes through.

3. **Isolation Forest (iForest) (Liu et al. 2008, 2012a):** iForest is a large number of binary tree-based unsupervised anomaly detection techniques. The methodology is founded on the principle of isolating trajectories, which means that abnormal trajectories are separated from the majority of other trajectories, whereas regular trajectories are endorsed by a substantial number of trajectory points.
4. **Extreme Gradient Boosting (XGBoost) (Chen and Guestrin 2016):** XGBoost is an efficient and flexible ensemble model. Its basic idea is to build a strong classifier with high accuracy through multiple simple classifiers. The trajectory features employed in our experiment include the relative positions between adjacent points, the angles between trajectories, and the durations of trajectories.
5. **ATD-RNN (Song et al. 2018):** ATD-RNN is an RNN-based anomaly detection model, including ATD-LSTM and ATD-GRU (the method for our experimental comparison), which captures trajectory temporal information and internal features between anomalous and normal trajectories through stacked RNN. To tackle the data sparsity issue, ATD-RNN considers a variety of starting points and end points, as well as structural properties from related trajectories.

Results and Analysis

We compared RETAD with the above baselines to verify its effectiveness and innovation. The test was completed on the synthetic anomalous trajectory data sets Traj-1, Traj-2..., Traj-6, and the evaluation metrics (i.e., ACC, P, R, F1) of different anomaly detection methods are depicted in Table 2.

As seen from Table 2, in the majority of cases, the evaluation metrics of our RETAD method are higher than the baselines, especially on the recall and F1 score. Compared with baselines, we make the following analysis:

Table 2. Comparison of Anomaly Detection Algorithms on Multiple Metrics

		LCSS	TOP-EYE	iForest	XGBoost	ATD-RNN	RETAD (ours)
Traj-1	ACC	0.8611	0.9356	0.8856	0.8978	0.9435	0.9661
	P	0.8265	0.9211	0.9003	0.8547	0.9276	0.9418
	R	0.7832	0.9078	0.7226	0.6821	0.9134	**0.9572**
	F1	0.8043	0.9144	0.8017	0.7587	0.9204	**0.9494**
Traj-2	ACC	0.8446	0.9136	0.8611	0.8376	0.9388	0.9703
	P	0.7812	0.8824	0.8523	0.8749	0.8993	0.9314
	R	0.7735	0.8730	0.7217	0.6908	0.9025	**0.9462**
	F1	0.7773	0.8777	0.7816	0.7720	0.9009	**0.9387**
Traj-3	ACC	0.8864	0.8947	0.8533	0.8473	0.9554	0.9566
	P	0.8215	0.8553	0.8321	0.7762	0.8986	0.9670
	R	0.7143	0.8652	0.7547	0.5433	**0.9674**	0.9588
	F1	0.7642	0.8602	0.7915	0.6392	0.9317	**0.9629**
Traj-4	ACC	0.9014	0.9060	0.8607	0.8828	0.9227	0.9431
	P	0.8617	0.8319	0.8282	0.9135	0.9732	0.9579
	R	0.7533	0.8537	0.8055	0.7289	0.9367	**0.9468**
	F1	0.8039	0.8427	0.8167	0.8108	**0.9546**	0.9523
Traj-5	ACC	0.8632	0.8928	0.8255	0.7928	0.9108	0.9336
	P	0.8025	0.8139	0.7918	0.8694	0.8932	0.8817
	R	0.8380	0.8605	0.6877	0.6012	0.8753	**0.9279**
	F1	0.8199	0.8366	0.7361	0.7108	0.8842	**0.9042**
Traj-6	ACC	0.8722	0.8769	0.8489	0.8633	0.8859	0.9261
	P	0.8137	0.8234	0.8104	0.8905	0.9147	0.9384
	R	0.7725	0.8026	0.7413	0.5778	0.8633	**0.8997**
	F1	0.7926	0.8129	0.7743	0.7009	0.8883	**0.9186**

Note: Accuracy (ACC), Precision (P), Recall (R), F1-score (F1). Longest Common Subsequence (LCSS), Isolation Forest (iForest), Extreme Gradient Boosting (XGBoost). The **bolded** values indicate the optimal results.

1. Compared with the traditional distance-based LCSS and machine learning methods, TOP-EYE, ATD-RNN, and our RETAD all make good use of historical trajectories, thus achieving relatively better detection results. In particular, the TOP-EYE method introduces decline functions to mitigate the effect of historical trajectories on evolutionary anomaly scores. Therefore, it is more competitive with LCSS and iForest as well as XGBoost. The dismal detection results of LCSS, iForest, and XGBoost may be because these approaches simply analyze the geometry and statistical features of trajectories, while disregarding the historical and sequential information.
2. The deep learning models ATD-RNN and RETAD achieve excellent performance on various metrics. Especially in terms of recall, it shows that the ATD-RNN and RETAD models detect more true anomalous trajectories than other baselines, and this also indicates that the RNN model can capture the internal feature differences of anomalous and normal trajectories. Furthermore, the superior results of ATD-RNN and RETAD also illustrate the importance of trajectory spatiotemporal information for anomaly detection.
3. Compared with the recent ATD-RNN model, our RETAD method has obvious advantages in terms of recall and F1, which indicates that the autoencoder-based reconstruction model can complete trajectory reconstruction and error comparison well. Meanwhile, the addition of vehicle information sharing and global interaction modeling in RETAD also leads to the superior performance of our method. It should be mentioned that the number of anomalous trajectories in the testing trajectory data gradually increases from Traj-1 to Traj-6, which verifies that the RETAD model can handle trajectory anomaly detection tasks in different complex scenarios.

CONCLUSION AND FUTURE WORK

In this paper, we propose an anomaly detection model RETAD based on trajectory reconstruction error. The model is based on the autoencoder, in which the encoder and decoder modules are implemented by GRU, and a module dealing with vehicle interactions is integrated into the autoencoder. The RETAD model is unsupervised learning, which can avoid the overfitting caused by unbalanced samples in trajectory anomaly detection, and the GRU-based autoencoder can effectively extract trajectory spatiotemporal features. Experimental results demonstrate that the effectiveness of our suggested RETAD in detecting anomalies is superior to traditional distance-based, density-based, and machine learning classification algorithms on multiple metrics, and it is also competitive with deep learning-based methods.

In future studies, we will further improve the anomalous trajectory data set and try to employ an attention mechanism, graph neural network, and other methods to expand the RETAD model to enhance the effectiveness of detecting trajectory anomalies. Simultaneously, applying the RETAD model to the trajectory data of other mobile agents (such as pedestrians, nonmotor vehicles, and animals) for anomaly detection is also a future research direction.

ACKNOWLEDGMENT

We are very grateful to Pengfei Liu and QiongYan Wang for their hard work in generating the testing data, and to Hang Yao for his constructive suggestions during the data processing.

FUNDING AGENCY INFORMATION

The work was jointly supported by the National Natural Science Foundations of China under Grant No. 62272364,61902296, the Province Key Research & Development Program of Shaanxi under Grant No. 2020LSFP3-15, Guangxi Key Laboratory of Trusted Software (No. KX202061), and the Key Research & Development Projects of Qingdao Science and Technology Plan under grant No.21-1-2-18-xx.

REFERENCES

Bouritsas, G., Daveas, S., Danelakis, A., & Thomopoulos, S. C. (2019). Automated real-time anomaly detection in human trajectories using sequence to sequence networks. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1–8). IEEE. doi:10.1109/AVSS.2019.8909844

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58. doi:10.1145/1541880.1541882

Chao, Q., Deng, Z., Ren, J., Ye, Q., & Jin, X. (2017). Realistic data-driven traffic flow animation using texture synthesis. *IEEE Transactions on Visualization and Computer Graphics*, *24*(2), 1167–1178. doi:10.1109/TVCG.2017.2648790 PMID:28092561

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. doi:10.1145/2939672.2939785

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the properties of neural machine translation: Encoder-decoder approaches*. 10.3115/v1/W14-4012

Fu, Q., Zhu, J., Chen, Y., Wan, J., & He, B. (2020). An automatic learning model for trajectory outlier detection. In *Proceedings of the 3rd International Conference on Data Science and Information Technology* (pp. 220–226). ACM. doi:10.1145/3414274.3414505

Ge, Y., Xiong, H., Zhou, Z. H., Ozdemir, H., Yu, J., & Lee, K. C. (2010). Top-eye: Top-k evolving trajectory outlier detection. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (pp. 1733–1736). ACM.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems* (pp. 2672-2680). MIT Press.

Hinton, G. E., & Zemel, R. (1993). Autoencoders, minimum description length and Helmholtz free energy. In *Proceedings of the 6th International Conference on Neural Information Processing Systems* (pp. 3-10). Morgan Kaufmann.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735 PMID:9377276

Huang, Y., & Zhang, Q. (2019). Identification of anomaly behavior of ships based on KNN and LOF combination algorithm. In AIP Conference Proceedings (vol. 2073, no. 1, p. 020090). AIP Publishing LLC. doi:10.1063/1.5090744

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in Statistics* (pp. 492–518). Springer. doi:10.1007/978-1-4612-4380-9_35

Kumar, D., Bezdek, J. C., Rajasegarar, S., Leckie, C., & Palaniswami, M. (2017). A visual-numeric approach to clustering and anomaly detection for trajectory data. *The Visual Computer*, *33*(3), 265–281. doi:10.1007/s00371-015-1192-x

Lee, J. G., Han, J., & Li, X. (2008). Trajectory outlier detection: A partition-and-detect framework. In *2008 IEEE 24th International Conference on Data Engineering* (pp. 140–149). IEEE. doi:10.1109/ICDE.2008.4497422

Li, X., Han, J., Kim, S., & Gonzalez, H. (2007). Roam: Rule-and motif-based anomaly detection in massive moving object data sets. In *Proceedings of the 2007 SIAM International Conference on Data Mining* (pp. 273–284). Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611972771.25

Liatsikou, M., Papadopoulos, S., Apostolidis, L., & Kompatsiaris, I. (2021). A denoising hybrid model for anomaly detection in trajectory sequences. EDBT/ICDT Workshops.

Lin, W., Zhou, Y., Xu, H., Yan, J., Xu, M., Wu, J., & Liu, Z. (2016). A tube-and-droplet-based approach for representing and analyzing motion trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(8), 1489–1503. doi:10.1109/TPAMI.2016.2608884 PMID:28113652

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). IEEE. doi:10.1109/ICDM.2008.17

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012a). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, *6*(1), 1–39. doi:10.1145/2133360.2133363

Liu, L., Qiao, S., Zhang, Y., & Hu, J. (2012b). An efficient outlying trajectories mining approach based on relative distance. *International Journal of Geographical Information Science*, *26*(10), 1789–1810. doi:10.1080/13658816.2012.654792

Liu, Y., Zhao, K., Cong, G., & Bao, Z. (2020). Online anomalous trajectory detection with deep generative sequence modeling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (pp. 949–960). IEEE.

Liu, Z., Pi, D., & Jiang, J. (2013). Density-based trajectory outlier detection algorithm. *Journal of Systems Engineering and Electronics*, *24*(2), 335–340.

Luan, F., Zhang, Y., Cao, K., & Li, Q. (2017). Based local density trajectory outlier detection with partition-and-detect framework. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 1708–1714). IEEE.

Lv, Z., Xu, J., Zhao, P., Liu, G., Zhao, L., & Zhou, X. (2017). Outlier trajectory detection: A trajectory analytics based approach. In *International Conference on Database Systems for Advanced Applications* (pp. 231–246). Springer.

Ma, C., Miao, Z., Li, M., Song, S., & Yang, M. H. (2018). Detecting anomalous trajectories via recurrent neural networks. In *Asian Conference on Computer Vision* (pp. 370–382). Springer.

Morris, B. T., & Trivedi, M. M. (2011). Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2287–2301.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32.

Piciarelli, C., Micheloni, C., & Foresti, G. L. (2008). Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1544–1554.

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. https://arxiv.org/abs/1609.04747

San Román, I., Martín de Diego, I., Conde, C., & Cabello, E. (2019). Outlier trajectory detection through a context-aware distance. *Pattern Analysis & Applications*, *22*(3), 831–839.

Song, L., Wang, R., Xiao, D., Han, X., Cai, Y., & Shi, C. (2018). Anomalous trajectory detection using recurrent neural network. In *International Conference on Advanced Data Mining and Applications* (pp. 263–277). Springer.

Tang, J., & Ngan, H. Y. (2016). Traffic outlier detection by density-based bounded local outlier factors. *Information Technology in Industry, 4*(1). 10.17762/itii.v4i1.38

Thompson, K. (2016). U.*S. department of transportation federal highway administration. next generation simulation (NGSIM) vehicle trajectories and supporting data* [Dataset]. Provided by ITS DataHub through Data.transportation.gov. 10.21949/1504477

Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., & Huang, G. B. (2017). Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology. *IEEE Transactions on Intelligent Transportation Systems*, *19*(5), 1559–1582.

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Proceedings 18th International Conference on Data Engineering* (pp. 673–684). IEEE.

Yang, W., Gao, Y., & Cao, L. (2013). TRASMIL: A local anomaly detection framework based on trajectory segmentation and multi-instance learning. *Computer Vision and Image Understanding*, *117*(10), 1273–1286.

Zhu, J., Jiang, W., Liu, A., Liu, G., & Zhao, L. (2017). Effective and efficient trajectory outlier detection based on time-dependent popular route. *World Wide Web (Bussum)*, *20*(1), 111–134.

Chaoneng Li is a Ph.D. candidate with School of Computer Science and Technology, Xidian University. Prior to that, he received his B.S. degree in the Internet of Things in 2017 from Northwest Normal University. Recently, his main research focuses on computer vision, anomaly detection, trajectory mining and traffic prediction.

Guanwen Feng received his B.S. degree in software engineering from Hangzhou Dianzi University in 2020. He is current a Ph.D. candidate with School of Computer Science and Technology, Xidian University. His research interests include text to speech, computational biology, traffic prediction, calligraphy font generation.

Yiran Jia received her B.S. degree in Computer Science and Technology from Anhui University of Science and Technology in 2020. She is now a master's student at the School of Computer Science and Technology, Xidian University. Her research interest covers automatic driving trajectory prediction and evaluation.

Yunan Li received his B.S. degree and the Ph.D. degree from Xidian University in 2014 and 2019. Since January 2020, he has been a faculty member in the School of Computer Science and Technology, Xidian University. His main research interests include computer vision and machine learning, especially for image enhancement and gesture/action recognition. He has published 14 papers as first author or first student author in top journals and conferences, such as IJCV, TCSVT, ICCV and AAAI. He has served as the reviewer on several top journals and conferences, such as IEEE TCSVT, TITS, CVPR, AAAI, ICME.

Jian Ji was born in Xi'an, China, in 1971. She received the B.Sc. degree in computational mathematics from Northwest University, Xi'an, in 1993, and the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, in 2007. She is currently a professor with the School of Computer Science and Technology, Xidian University, Xi'an. Her research interests include computational intelligence, pattern recognition, and image analysis.

Qiguang Miao received his Ph.D. degree in computer application technology from Xidian University, in 2005. He is currently a Professor and a Ph.D. Student Supervisor with the School of Computer Science and Technology, Xidian University. His research interests include machine learning, intelligent image processing, and malware behavior analysis. As the Principal Investigator, he is conducting or has completed four projects through the NSFC, two projects through the Shaanxi Provincial Natural Science Fund, over ten projects through the National Defence Preresearch Foundation, and 863 projects through the Weapons and Equipment Fund. He has hosted one project supported by the Fundamental Research Funds for the Central Universities by MOE. In recent years, he has published over 100 papers in significant domestic and international journals or conferences, including the IEEE Transactions on Image Processing, the IEEE Transactions on Geoscience and Remote Sensing, the Journal of Visual Communication and Image Representation, NeuroComputing, and the IET Image Processing, Knowledge Based Systems, 30 of which are indexed by SCI, and over 40 of which are included in EI. He is a Committee Member of the Editorial Board of the Internet of Things and the Assessment Expert of the State Science and Technology Prizes and the National Defense Basic Scientific Research Project.