Mathematical Biosciences and Engineering

MBE, 20(8): 15075–15093.
DOI: 10.3934/mbe.2023675
Received: 15 February 2023
Revised: 27 June 2023
Accepted: 09 July 2023
Published: 14 July 2023

http://www.aimspress.com/journal/MBE

Research article

A trajectory outlier detection method based on variational auto-encoder

Longmei Zhang[1], Wei Lu[2,*], Feng Xue[2] and Yanshuo Chang[2]

[1] School of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
[2] School of Information, Xi'an University of Finance and Economics, Xi'an 710100, China

* **Correspondence:** Email: 908220526@qq.com.

Abstract: Trajectory outlier detection can identify abnormal phenomena from a large number of trajectory data, which is helpful to discover or predict potential traffic risks. In this work, we proposed a trajectory outlier detection model based on variational auto-encoder. First, the model encodes the trajectory data as parameters of distribution functions based on the statistical characteristics of urban traffic. Then, an auto-encoder network is built and trained. The training goal of the auto-encoder network is to maximize the generation probability of original trajectories when decoding. Once the model training is completed, we can detect the trajectory outlier by the difference between a trajectory and the trajectory generated by the model. The advantage of the proposed model is that it only needs to compute the difference between the original trajectory and the trajectory generated by the model when detecting the trajectory outlier, which greatly reduces the amount of calculation and makes the model very suitable for real-time detection scenarios. In addition, the distance threshold between the abnormal trajectory and the normal trajectory can be set by referring to the proportion of the abnormal trajectory in the training data set, which eliminates the difficulty of setting the threshold manually and makes the model more convenient to be applied in different actual scenes. In terms of effect, the proposed model has achieved more than 95% in accuracy, which is better than the two typical density-based and classification-based detection methods, and also better than the methods based on machine learning in recent years. In terms of efficiency, the model has good convergence in the training phase and the training time increases slowly with the data scale, which is better than or as the same as the comparison methods.

Keywords: trajectory outlier detection; machine learning; variational auto-encoder; trajectory similarity

1. Introduction

With the popularization of various mobile terminals with high-precision positioning function, such as automobile navigation, mobile phones and wearable devices, a large number of trajectory data arose. These trajectory data contain abundant spatiotemporal and semantic information, which have been proved to be very valuable resources and have a wide range of application scenarios [1]. Taxi is a very important urban transportation tool. All taxis in operation frequently report their location information to the data center through installed GPS or Beidou positioning equipment. These massive trajectories data are widely distributed in the urban road network, which can well reflect the traffic conditions of the urban road network. Therefore, taxis are also known as the "flow detector" of urban traffic. The taxi trajectory data set is also widely used in urban planning, road recommendation, traffic hotspot analysis, traffic accident analysis and other fields [2].

The purpose of trajectory outlier detection is to identify abnormal phenomena from a large number of trajectory data, which is helpful to discover or predict potential traffic risks [3]. The trajectory outlier is generally divided into two categories: 1) The trajectory deviates from other trajectories in space; 2) The time sequence of trajectory points is significantly different from other trajectories. Trajectory outlier means that some behaviors deviate from expectations, usually accompanied by special or interesting events. Taking urban traffic monitoring and management as an example, in the saturated urban road network, local trajectory outliers may reflect the occurrence of traffic accidents, bad weather or road emergencies. Detection and real-time analysis of these trajectory outliers can not only provide the traffic management department with the basis for traffic situation analysis, timely find and predict the congested sections, but also provide the driver with reference for route selection.

Traditional trajectory outlier detection methods mainly focus on trajectory similarity and clustering analysis [4]. These methods calculate the similarity between trajectories by defining the distance metric, and then divide the trajectories into several clusters by clustering. Finally, the cluster with a small number of trajectories is identified as outliers by threshold setting. However, the efficiency of such methods is usually not high, and it cannot well meet the needs of real-time analysis of large-scale trajectory data. In the field of urban transportation, the trajectory data has the characteristics of uncertainty, sparsity, skewed distribution and continuous updating, which makes it more difficult to detect outliers in real-time [5]. With the rise of deep learning methods, deep neural networks have gradually been widely used in trajectory data analysis and mining [6]. Aiming at the problem of online real-time detection of trajectory outliers, we proposed a trajectory outlier detection model based on variational auto-encoder in this paper. The model encodes the trajectory data as parameters of distribution functions and an auto-encoder network is built and trained. Once the model training is completed, we can detect the trajectory outlier by the difference between a trajectory and the trajectory generated by the model. The main contributions include:

1) A deep learning networks based on variational auto-encoder is used to learn the trajectory characteristics and distribution. Once the model training is completed, we can detect the trajectory outlier by the difference between a trajectory and the trajectory generated by the model. Compared with the traditional density-based or classification-based trajectory outlier detection method, the calculation amount is greatly reduced which makes the model very suitable for real-time detection scenarios. Compared to other methods using variational auto-encoder, this paper performs segmentation and differential processing on trajectory data for input of the model, which make it

unnecessary to eliminate noise points in trajectories and can further improve the accuracy of the model.

2) When comparing an original trajectory with the trajectory generated by the model, the difference between them is calculated from two aspects of Euclidean distance and cosine similarity, which further improve the detection effect of outliers.

3) When detecting the trajectory outliers, the distance threshold between the outliers and the normal trajectories can be set by referring to the proportion of the abnormal trajectory in the training data set, which eliminates the difficulty of setting the threshold manually and makes the model more convenient to be applied in different actual scenes.

In the remainder of this paper, we introduce related work in Section 2. After that, we give some definitions formally in Section 3. Section 4 introduces our model and methods for outlier trajectory detection. An experimental evaluation and analysis of the effectiveness and efficiency of our methods is presented in Section 5. Finally, we conclude the work and briefly discusses future work in Section 6.

2. Related work

Trajectory outlier detection methods mainly include density-based methods [7–9], classification-based methods [10–12], grid-based methods [13–15] and deep learning-based methods [16].

The density-based detection method uses algorithms such as k-means [17], DBSCAN [18] or improved DBSCAN [19] to cluster the segmented trajectories, and the sparse trajectories are determined as outliers. The commonly used definitions of distance between trajectories in these algorithms include European distance, DTW distance [20], LCSS distance [21] and Hausdorff distance [22]. Trajectory similarity calculation is the core of this kind of method. Recently, D. Zhang et al. studied this problem and proposed a method suitable for continuous calculation of trajectory similarity, and tried to apply this method to trajectory outlier detection [23]. In practical application, threshold selection is one of the difficult problems in this kind of methods. In addition, the large amount of computation makes these methods cannot meet the needs of real-time detection for large-scale trajectory data.

The idea of classification-based detection method is to establish a classifier model and train the model through the labeled trajectory data set so as to obtain normal trajectory features. Once the model training is completed, the trajectory outlier detection efficiency is very high. However, the disadvantage of this method is that it requires label data, and data annotation usually leads to a large amount of labor and time costs.

The main idea of the grid-based detection method is to map the trajectory data to the urban map grid after preprocessing, and then transform the detection of trajectory outliers into the detection of abnormal grid cell symbol sequence. This method is mainly applied to the detection of trajectory outliers in urban fixed road network environment.

In recent years, deep learning [24] developed rapidly and has attracted the attention of a large number of researchers. It has been widely used in almost all research fields related to feature extraction. This method has also attracted more and more researchers' attention in the field of trajectory data mining. Compared with the traditional method, the deep neural network can automatically learn the feature of the trajectory from the data set, and can achieve better results when the data set is sufficient. Document [25] proposes a pedestrian trajectory prediction method with dual attention fusion mechanism, which can also be used as an abnormal event detection method. In view of the problem of trajectory outlier detection caused by taxi fraud or detour, A. Belhadi et al. proposed a novel hybrid framework and two phase-based algorithms to identify trajectory outliers [26].

Auto-encoder (AE) [27] is a very important unsupervised learning method in deep learning. The auto-encoder is composed of an encoder and a decoder, which aims to learn the effective information from a large number of unmarked data, and realize the nonlinear compression and reconstruction of the input data. The goal of the traditional auto-encoder is to make the output and input as same as possible, but in the actual application, what we really care about is the hidden layer expression, so there are many improvement methods for the auto-encoder. The variational auto-encoder (VAE) [28] is one of the improved methods for auto-encoder. This method combines the deep learning method with Bayes on the basis of maintaining the basic function of the auto-encoder. It can be well applied to data generation [29] and anomaly detection in the case of no label data. Therefore, some researchers try to apply the VAE method to the field of trajectory outlier detection in different scenarios [30–33].

3. Definitions for distances and VAE

3.1. Trajectory and distance

Trajectory is an important spatiotemporal data type, which represents the information history of the state of moving objects changing continuously with time. The trajectory can be regarded as a mapping from time to state. i.e., F: $R^+ \rightarrow S^d$, where d is the dimension of the state space. For trajectory outlier detection, some preliminary definitions are discussed below.

Definition 1: A trajectory T is a sequence of time-ordered points, denoted by $T = (p_0, p_2, \ldots, p_i, \ldots, p_{n-1})$, where $p_i \in \mathbb{R}^2$ is the physical location (i.e., latitude and longitude), denoted by $p_i = (lon_i, lat_i)$, p_1 and p_{n-1} are the start and end points of the trajectory respectively.

Definition 2: A trajectory segment refers to the segment formed by the connection of adjacent points in the trajectory, denoted by $seg_i(T) = (s_i, e_i)$, where s_i and is e_i are the start and end points of the segment.

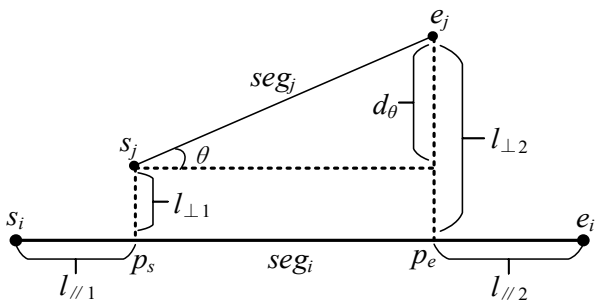

Figure 1. Space distance between trajectory segments.

Typically, measurement of distance between two trajectory segments include vertical distance ($d_\perp$), parallel distance($d_{/\!/}$) and angular distance(d_θ) [18]. For two trajectory segments $seg_i(T) = (s_i, e_i)$ and $seg_j(T) = (s_j, e_j)$, the different distances between them are defined in Eqs (1)–(3), as shown in Figure 1.

$$d_\perp(seg_i, seg_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}} \tag{1}$$

$$d_{\mathrm{P}}(seg_i, seg_j) = \min(l_{\mathrm{P1}}, l_{\mathrm{P2}}) \tag{2}$$

$$d_{\theta}(seg_i, seg_j) = \begin{cases} \left\| seg_j \right\| \times \sin\theta,\ 0 \le \theta < 90^{\mathrm{o}} \\ \left\| seg_j \right\|, 90^{\mathrm{o}} \le \theta < 180^{\mathrm{o}} \end{cases} \tag{3}$$

3.2. Variational auto-encoder

Auto-encoder is a neural network that tries to copy the input to the output, that is, reproduce the original data as much as possible. The auto-encoder includes two parts: encoder and decoder.

The encoder converts the input signal into a hidden layer expression through a certain mapping to learn the characteristics of the data, as shown in Eq (4). The decoder tries to remap the hidden layer expression into the original input signal through the learned feature expression, as shown in Eq (5).

$$z = \sigma_e(W_1 T + b_1) \tag{4}$$

$$T' = \sigma_d(W_2 z + b_2) \tag{5}$$

T and T' represent the trajectory input and output space, W_1 and b_1 are the weights and offsets of the encoding stage, W_2 and b_2 are the weights and offsets of the decoding stage, σ_e and σ_d are the nonlinear transformations. The objective of auto-encoder optimization is to minimize the error between T and T' as much as possible.

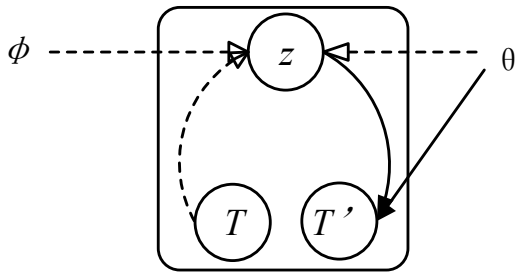

Figure 2. Variational auto-encoder probability model.

Variational auto-encoder (VAE) is an improved model of auto-encoder proposed by Kingma et al. [16] in 2014, which is mainly used for data generation. The basic idea of VAE is to assume that all data are generated by statistical process, and the distribution characteristics of data should be considered in the process of encoding and decoding. Therefore, the difference between VAE and traditional auto-encoder is that the auto-encoder compresses the input data into a fixed code in the hidden space, while VAE converts the input data into parameters of statistical distribution function, i.e., mean and standard deviation (μ, σ). Finally, the hidden space distribution parameters are optimized through network training to maximize the generation probability of the original input data during decoding. The basic principle of VAE is shown in Figure 2.

In this paper, we reconstruct the trajectory through VAE. The VAE model is trained with the actual trajectory data set T, and finally the output trajectory data T' is generated by the hidden variable Z. $T \rightarrow Z$ is the feature extraction and recognition model $q_\phi(Z \mid T)$, which is completed by the encoding process of the auto-encoder. $Z \rightarrow T'$ is the generation model $p_\theta(T' \mid Z)$, which is completed by the decoding process of the auto-encoder. Assuming that the trajectory data $T = [T_1, T_2, \ldots, T_N]$ are all independent and identically distributed, the maximum likelihood method is used for parameter estimation, as shown in Eq (6).

$$\log P_\theta(T_1, T_2, ..., T_N) = \sum_{i=1}^{N} \log P_\theta(T_i) \tag{6}$$

VAE uses feature extraction and recognition model $q_\phi(Z \mid T_i)$ to approximate the real posterior probability $p_\theta(Z \mid T_i)$, and uses KL divergence to measure the similarity of the two distributions, as shown in Eq (7).

$$\begin{aligned} KL(q_\phi(Z \mid T_i) \,\|\, p_\theta(Z \mid T_i)) &= E_{q_\phi(Z|T_i)} \log \frac{q_\phi(Z \mid T_i)}{p_\theta(Z \mid T_i)} \\ &= E_{q_\phi(Z|T_i)} (\log \frac{q_\phi(Z \mid T_i)}{p_\theta(Z, T_i)} + \log p_\theta(T_i)) \end{aligned} \tag{7}$$

Assuming that the sample input conforms to the normal distribution, we set two encoders in VAE, one for calculating the mean and the other for calculating the variance. In the mean value calculation network, the robustness of the result is improved by adding "Gaussian noise", and the encoder is regularized by KL loss to ensure that the encoder result has zero mean value. The variance calculation network is used to dynamically adjust the noise intensity. When the reconstruction result error is large (greater than the KL threshold), the noise is appropriately reduced. When the reconstruction result error is small (less than the KL threshold), the noise is appropriately increased and the generation ability of the decoder is improved through training. The essence of VAE is to find a suitable probability distribution parameter $\theta = (\mu, \sigma)$ for each input X_i through continuous training. The overall neural network structure of VAE is shown in Figure 3.

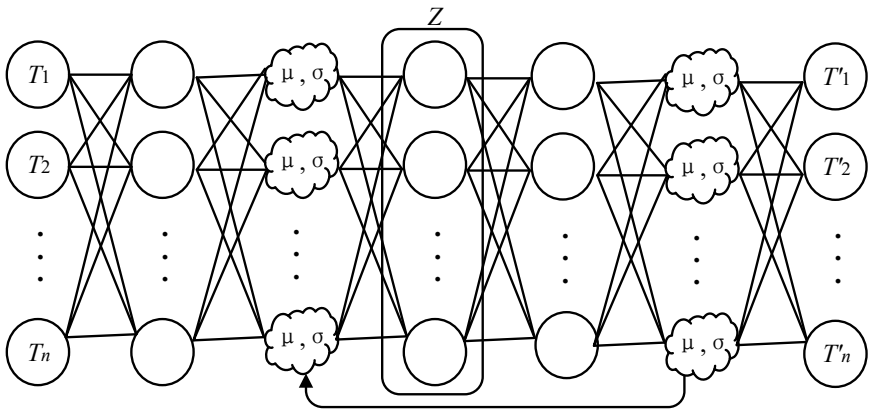

Figure 3. The neural network architecture of VAE.

4. Trajectory outlier detection method based on variational auto-encoder

4.1. Model

Normal trajectories are generally continuous and smooth, so their trajectories reconstructed by VAE should maintain good consistency with the original trajectories. Abnormal trajectories are generally not smooth, usually manifested as position drift, sharp changes in speed or motion direction, so their trajectories reconstructed by VAE should be greatly different from the original trajectories. Under the guidance of this basic idea, a trajectory outlier detection model based on variational auto-encoder is proposed. The model consists of three parts: the trajectory data preprocessing module, the trajectory data generation module and the trajectory outlier determination module, as shown in Figure 4.

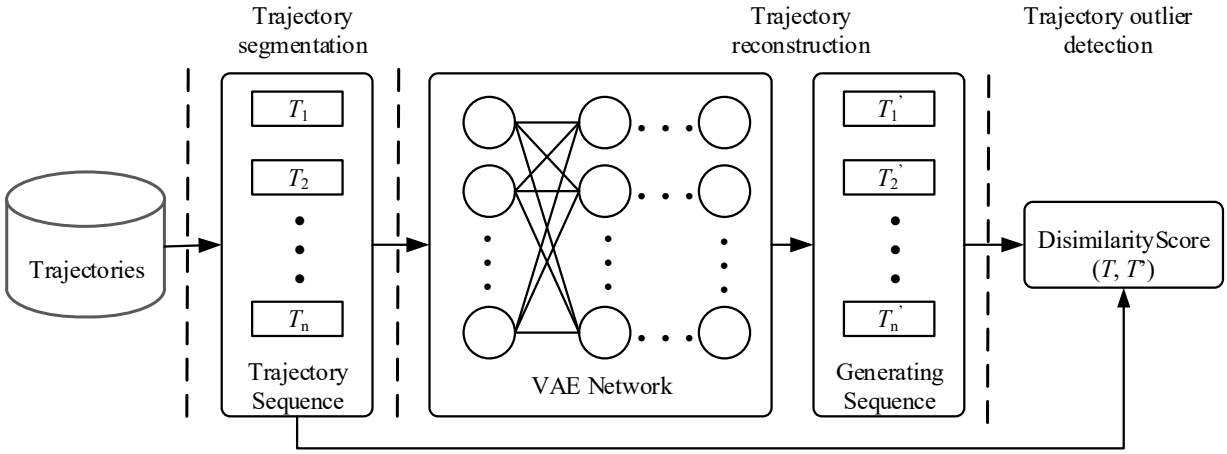

Figure 4. Trajectory outlier detection model based on VAE.

In this paper, Long Short-Term Memory (LSTM) network is used as the basic unit of encoder and decoder in VAE Network. LSTM is a classical sequence modeling network. It solves the problem of long sequence dependence through forgetting gate, control gate and output gate. The current output is determined according to the output at the previous time and the input at this time. The basic structure of LSTM we used in this paper is shown in Figure 5.

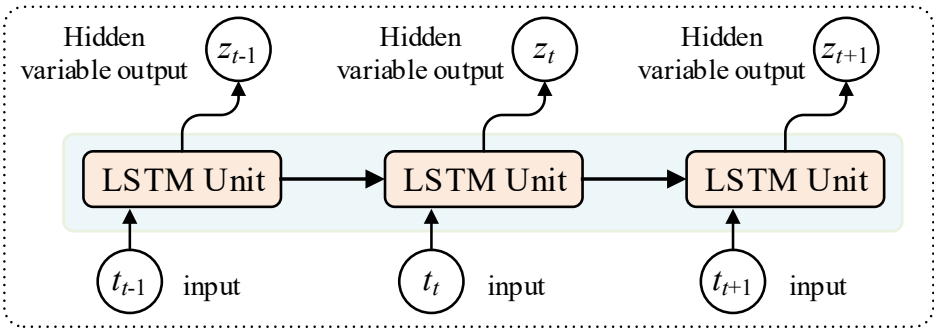

Figure 5. Basic LSTM structure of Encoder/Decoder.

4.1.1. Preprocessing of trajectory data

In general, the trajectories in the dataset are sequences of different lengths, while the input format required by the VAE is equal length sequence. Therefore, it is necessary to divide the trajectories in the data set into equal length sub trajectories first. In this paper, the sliding window method is used to process the original trajectory, which not only guarantees the integrity of the original trajectory, but also preserves the dependence of the trajectory points on the time series.

Let N multi-dimensional trajectory sequence samples $T = [T_1, T_2, \ldots, T_N]$, where a single sample with length M is represented as $T^n = (p_0^n, p_1^n, ..., p_i^n, ..., p_{M-1}^n)$, and p_i^n represents the trajectory point at the ith time of the nth sample. We use longitude and latitude data to represent a trajectory point. When the sliding window size is w and the time step is s, the sample T^n is segmented as shown in Eq (8).

$$T^{sn} = f(T^n) = \begin{bmatrix} T_1^n \\ T_2^n \\ \mathrm{L} \\ T_i^n \\ \mathrm{L} \\ T_d^n \end{bmatrix} = \begin{bmatrix} \left[p_0^n, \mathrm{L}\ , p_{w-1}^n \right] \\ \left[p_s^n, \mathrm{L}\ , p_{s+w-1}^n \right] \\ \mathrm{L} \\ \left[p_{(i-1)s}^n, \mathrm{L}\ , p_{(\mathrm{i}-1)s+w-1}^n \right] \\ \mathrm{L} \\ \left[p_{(d-1)s}^n, \mathrm{L}\ , p_{M-1}^n \right] \end{bmatrix} \tag{8}$$

After the actual trajectory data set is processed by sliding window, the equal length sub trajectory set can be obtained, which meets the equal length requirement of VAE network for input sequence. However, the spatial position coordinates of the trajectory points in the sub trajectory set do not meet the distribution consistency, which will cause the model to be biased. Therefore, the first-order difference method is used to further process the equal length sub trajectories. By calculating the first-order difference of the coordinates of the adjacent trajectory points in the trajectory segment, the position deviation P of the trajectory points at each time can be obtained and then the coordinate data of the trajectory points can be limited to [-max(P), +max(P)], which can ensure the consistence of the distribution of the point data.

4.1.2. VAE network training

The input for VAE network training is the trajectory data set of equal length and equal distribution after preprocessing, and the output is the distribution parameters of encoder and decoder. In the encoding stage, the parameters describing the distribution of each dimension in the hidden space are output through the neural network. Assuming that the trajectory data conforms to the normal distribution a priori, the mean and variance describing the distribution of the hidden state are output. In the decoding stage, the reparameterization trick is used to combine the mean and variance of the output in the encoding stage, sample from the standard normal distribution, generate the hidden variables through the neural network and reconstruct the original input and measure the distribution similarity through the divergence of Eq (7). The specific process of VAE network training is shown in Algorithm 1. Where, ϕ and θ represent parameters of Gaussian distribution of encoder and decoder respectively. In the process of model training, the learning rate is set to 0.005 and the convergence condition is that the training error is less than 0.01.

Algorithm 1 VAE model training

Input: T is the trajectory dataset, N is the size of dataset

Output: probabilistic encoder, probabilistic decoder

$\phi,\theta \leftarrow$ initialization parameter

repeat

  for i=1 to N do

    get L samples from ε: $N(0,1)$

    $z^{(i,l)} = h_\phi(\varepsilon^{(i)}, t^{(i)}),\ \ i = 1,...,N$ // Generate hidden variables

  end for

  $\mathrm{E}=\sum^{N} -KL(q_\phi(z \mid t^{(i)}) \| p_\theta(z)) + \frac{1}{T}\sum^{L}(\log p_\theta(t^{(i)} \mid z^{(i,l)}))$

  ϕ,θ $\leftarrow$ use gradients of E to update parameter

until ϕ,θ parameter convergence

4.1.3. Trajectory outlier detection

After the training, the VAE network acquired the trajectory point distribution characteristics. For normal trajectories, the trajectories reconstructed by VAE should be consistent with the original trajectories. Therefore, outliers can be found through the differences between the original trajectories and the reconstructed trajectories. For moving objects or vehicles, sudden change in speed or direction during driving means abnormal occurrence. Considering the spatial distance, direction and time factors, the difference between the original trajectory and the reconstructed trajectory is expressed as the vertical distance ($d_\perp$), the parallel distance ($d_{/\!/}$) and the angular distance (d_θ), as shown in Eq (9).

$$diff(seg_i, seg_j) = \alpha_1 d_\perp(seg_i, seg_j) + \alpha_2 d_{\mathrm{P}}(seg_i, seg_j) + \alpha_3 d_\theta(seg_i, seg_j) \tag{9}$$

α_1, α_2, α_3 are weight coefficients of vertical distance, parallel distance and angular distance respectively, $\alpha_1 + \alpha_2 + \alpha_3 = 1$. If the difference between the original trajectory and the generated trajectory is greater than a certain threshold τ, the trajectory should be determined to be an outlier. The selection of threshold τ has an important impact on the detection results. In general, the threshold τ should be customized according to the experience in different scenarios, which is a difficult problem in practical application. In order to solve this problem, this paper introduces the ratio parameter λ of abnormal trajectory, and adjusts the threshold τ adaptively through the parameter λ to meet the actual application needs. Step 1, the trajectory data set to be detected is used as input to train the VAE model using Algorithm 1 to obtain the distribution parameters of encoder and decoder. Step 2, take each trajectory in the trajectory dataset to be detected as input, and use the decoder to regenerate the trajectory to obtain the generated trajectory set. Step 3, Calculate the distance between each trajectory in the validation dataset and its generated trajectory, and sort it in descending order. Step 4, Use the ratio λ of abnormal trajectories in the validation dataset as a condition to intercept the distance set and obtain the distance threshold τ of abnormal trajectories. Step 5, Determine whether each trajectory in the detected trajectory dataset is an abnormal trajectory based on the distance threshold τ. The specific trajectory outlier detection process is shown in Algorithm 2.

Algorithm 2 trajectory outlier detection

Input: T^N is the trajectory dataset for detecting, T^V is the validation dataset, λ is the abnormal proportion in T^V, α_1, α_2, α_3 are distance coefficients

Output: T^D is the abnormal trajectory data set detected

```
Encoder, Decoder ← VAEModelTrain (T^N, size(T^N))
refactoringSet ← Decoder.predict (T^V)
difference ←diff (T^V, refactoringSet, α1, α2, α3)
τ ← reverseOrder (difference) [size(T^V)*λ]
for t in T^N do
    if (diff (Decoder.predict (t), t) > τ)
        T^D.append(t)
    end if
end for
return T^D
```

5. Experiments and analysis

5.1. Experimental environment and dataset processing

The hardware platform of this experiment is CPU Intel (R) core (TM) i7-7700, 12 G memory, GPU Intel (R) HD graphics 630. The software platform of the experiment is operating system Windows10, machine learning framework Keras and algorithm implementation language Python.

The experimental data set is the public data set of taxis from Porto, Portugal, from early July 2013 to the end of June 2014 (https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data). The data set contains 1,710,670 tracks of 442 taxis in the whole year. The sampling frequency of trajectory points is 15 seconds, and some trajectories are shown in Figure 6.

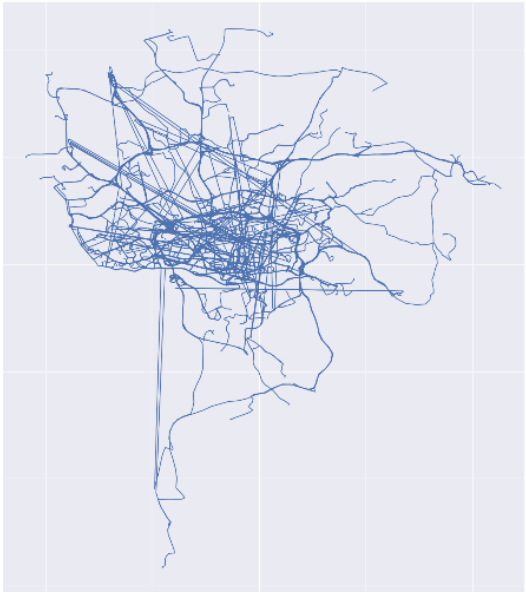

Figure 6. Partial trajectories of experimental data set.

First, the data set is filtered to remove the discontinuous trajectories, and then the sliding window method is used to acquire trajectory segments with equal length. The window size is set to $w = 25$ and the step size is set to $s = 1$. Then, select the verification set for labeling, and divide the trajectories into normal trajectories and abnormal trajectories. The criterion for judging whether the trajectory is normal or not is whether it includes position drift, sharp turn or rapid acceleration. Figure 7 shows several typical normal and abnormal trajectories. Figure 7(a) shows the situation where the trajectory deviates from most normal trajectories, as such sharp turns cannot occur on normal trajectories, and even exceed the limits of taxi turning operation. Figure 7(b) shows time sequence of trajectory points is significantly different from normal trajectories or there may be abnormal acceleration in the trajectory. Finally, through the first-order difference processing and normalization processing, the regular trajectory data set is obtained for model training and verification.

(a) Sharp turn (b) Rapid acceleration (c) Normal

Figure 7. Typical normal and abnormal trajectories.

5.2. Evaluation metrics

In this paper, precision, recall and F1 values are selected to evaluate the effect of the method we proposed. The calculation methods of the three indicators are shown in Eqs (10)–(12).

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

TP represents the number of outliers correctly identified as abnormal trajectories, *FN* represents the number of outliers incorrectly identified as normal trajectories, *FP* represents the number of normal trajectories incorrectly identified as outliers and *TN* represents the number of normal trajectories correctly identified as normal trajectories. In addition, the AUC value of ROC curve is used to evaluate the comprehensive recognition ability of the model for positive and negative samples.

5.3. Analysis of model generalization effect

5.3.1. Influence of training parameter epoch

In the experiment, 2000 trajectories were randomly selected from the preprocessed trajectory data set as the training set, and 500 trajectories were selected as the verification set, which including 400 normal tracks and 100 outliers. The threshold value λ in the trajectory outlier detection algorithm is set to 0.2. When the training parameter epochs = [200, 1000, 2000, 5000, 10,000], the outlier detection experiment was carried out separately. The effect of the model on each evaluation index is shown in Figure 8.

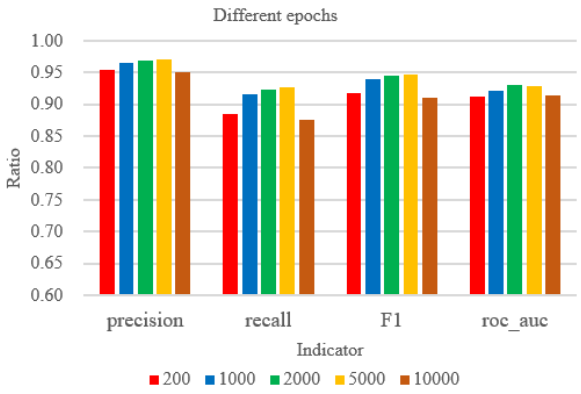

Figure 8. Model effect under different epochs.

Overall, the model has a high recognition effect, and the AUC value of the ROC curve always remains above 0.90. The difference of model effect is small under different training epochs, which reflects that the model has good convergence and stability. With the increase of the training epoch, the precision, recall and F1 value will increase first and then decrease slightly. The descending process shows that the trajectory reconstruction model may have a certain overfitting phenomenon in the process of learning the features. However, the problem of overfitting is not serious.

5.3.2. Influence of training set size train_size

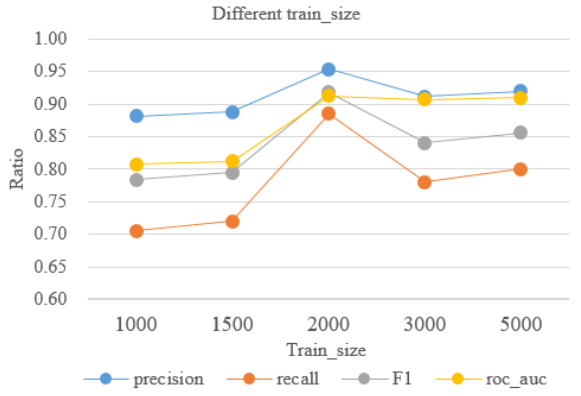

Figure 9. Model effect under different train_size.

This section analyzes the influence of the training set size on the model effect. Under the same verification set conditions as the previous section, set epochs = 200 and set the training set size parameter train_size = [1000, 1500, 2000, 3000, 5000], respectively carry out experiments on the model, and the performance of each evaluation index of the model is shown in Figure 9.

From the experimental results, it can be seen that with the increase of the training set size, the trajectory reconstruction model learns more and more fully the normal trajectory distribution features, and the recognition effect of outliers is also improved. When the training set size is increased to a certain extent, the model effect enters a relatively stable convergence state. In addition, the model is not very sensitive to the size of the training set, and it can still obtain good results even when the training set is small (The prerequisite is that the trajectory distribution characteristics are sufficient). For the experimental data set in this paper, when the size of the training set reaches about 2000, the model can achieve good results and enter the convergence state.

5.3.3. Influence of threshold λ in outlier detection algorithm

In this paper, in order to solve the difficult problem of manually setting the distance threshold when judging the outliers, the distance threshold is automatically adjusted by the parameter λ, which is the ratio of the abnormal trajectory. This section analyzes the effect of parameter λ on the trajectory outlier detection algorithm through experiments.

We first get the trajectory generation model when the training epochs = 200 and the training set size train_size = 2000. 500 trajectories including 86 abnormal trajectories are randomly selected from the trajectory data set as the verification set, which the ratio is 0.172. When different thresholds λ = [0.10, 0.15, 0.20, 0.25, 0.40, 0.50] are set in the trajectory outlier detection algorithm, the effect of the algorithm is shown in Figure 10.

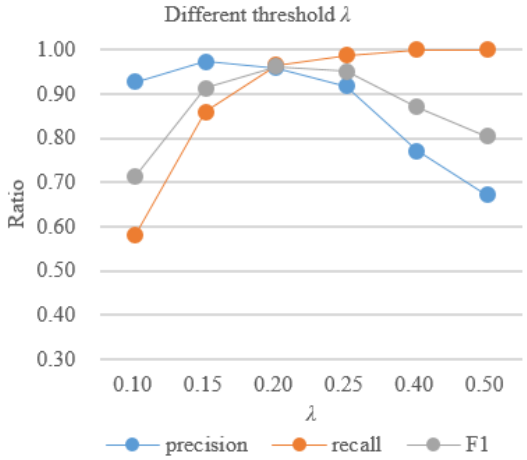

Figure 10. Model effect under different threshold λ.

From the experimental results, it can be seen that when the threshold λ of the trajectory outlier detection algorithm is set between 0.15 and 0.25, the algorithm achieves good results and the recall, precision and F1 value all remain at a high level. When λ is set to 0.2, the effect of the algorithm is optimal, and the recall, precision and F1 value are all kept above 95%. When λ is set to be greater than

or equal to 0.4, the recall rate will reach 100%, but the error rate will increase, which will lead to a decrease in the accuracy rate. The experimental results show that the algorithm achieves good results when the value of λ is near ratio of abnormal trajectory in the verification set. From this, we can learn that in practical application, if we pursue high precision, we can obtain the approximate proportion of outliers according to the samples in the data set, so as to provide a reference for setting the parameter λ of the algorithm. According to different application requirements, if we pursue higher recall, we can appropriately increase the λ value in the trajectory outlier detection algorithm.

5.4. Effect comparison

We select three typical types of methods as benchmarks. One is the density-based method, one is the classification-based method and the other is the deep learning based method.

1) Density-based methods: DTW distance-based clustering and Hausdorff distance based clustering.
2) Classification-based methods: KNN and deep clustering method [3].
3) Deep learning based methods: Sequence autoencoder (SAE) model [32] and GM-VSAE model [33].

For density-based method, we implement the two methods using the DBSCAN algorithm interface provided by Scikit learn library. The distance threshold value T_0 for the trajectory outlier of the clustering algorithm is 0.1, 0.3, 0.5, 0.7 and 0.9 respectively, and the quantity threshold value T_1 is 3, 5, 7, 9 and 11 respectively. The results of the optimal accuracy are used as the comparison benchmark.

For KNN and deep clustering methods, we implement the two methods based on literature [3]. The public data set of taxis from Porto, Portugal is used in literature [3] which is the same as that in this paper.

For deep learning-based method, we implement SAE and GM-VSAE algorithms based on literature [33]. The basic data sets used in the experiment are all also from the public data set of taxis from Porto, Portugal as the same as this paper. Literature [33] mainly focuses on two types of anomaly detection: detour and route switching. Therefore, in literature [33], drift points and abnormal turning points in trajectories are eliminated in data set preprocessing and then two different perturbation schemes are used to generate two types of anomalous trajectories. This paper does not distinguish between different types of abnormal trajectories, so it is not necessary to eliminate these abnormal points in the data preprocessing stage. On the contrary, we think that these outliers are very important and may reflect the occurrence of some special events. For SAE and GM-VSAE models, we take the best results under different parameter conditions for comparison.

Table 1. Test results of different methods.

Methods	Precision	Recall	F1	PR-AUC
DTW	0.649	0.726	0.685	0.711
Hausdorff	0.876	0.953	0.916	0.836
KNN [3]	0.713	0.601	0.656	0.64
Deep clustering [3]	0.862	0.783	0.821	0.84
SAE [32]	0.812	0.884	0.846	0.829
GM-VSAE [33]	0.832	0.941	0.883	0.871
VAE (ours)	**0.960**	**0.965**	**0.963**	**0.943**

The comparison indexes are still precision, recall, F1 value and PR-AUC value. The experimental results of trajectory outlier detection by different methods are shown in Table 1.

From the experimental results in Table 1, it can be seen that the trajectory outlier detection method based on variational auto-encoder VAE proposed in this paper is superior to the reference methods in all indicators.

The DTW based clustering method can coordinate the time alignment between trajectory points, and can achieve better results for clustering of unequal length trajectories. In this paper, the influence of unequal length trajectories is eliminated through data preprocessing. Therefore, the method based on DTW clustering cannot show advantages. The data set selected in this paper is the all-weather trajectory data of taxis. There is a large speed difference between the trajectory points in different periods. The clustering method based on Hausdorff distance can locally optimize the dislocation alignment between the trajectory points, and has a certain coordination for the speed difference between trajectory points. It can achieve good results when used in the clustering of urban vehicle trajectories in different periods in theory. However, the clustering method based on Hausdorff distance is sensitive to local outliers, which will increase the false positive rate to a certain extent. In the experimental results, the precision of the clustering method based on Hausdorff distance is lower than the recall rate, which also shows this. KNN also needs to calculate the distance in the clustering process, so the effect is lower than DTW and Hausdorff which are based on density clustering. Deep clustering method trained a binary pairwise deep neural network to cluster the sequence of trajectory represented as trips. Dynamic Time Warping (DTW) is used to calculate the distance between two ordered degree sequences. This method has achieved good results, but road network information needs to be used in trajectory data preprocessing.

The PR-AUC value of method SEA and method GM-VSAE exceeds 0.8 under the best parameters, which is a good result. However, the premise of such a good result is that it is oriented to specific types of anomalies diagnosis. When these two methods are used to diagnose another type of anomalies, they may not be suitable. For example, when these two methods are used to diagnose route switching anomalies, the PR-AUC value drops to about 0.7.

In this paper, we use the VAE model to obtain the distribution characteristics of trajectory points, which can also eliminate the influence of velocity difference between different trajectories through trajectory reconstruction. It is also less affected by local outliers, so it performs better in accuracy. In addition, the method in this paper obtains the distribution characteristics of normal trajectory points through the first-order difference method in the learning stage, and it is not necessary to distinguish different types of anomalies in the diagnosis stage.

5.5. Efficiency analysis

In practical application, when the trajectory outlier detection method achieves more than 90% recall and precision, it will have a good usability. In the field of urban traffic management under the background of the current big data era, the demand for real-time trajectory outlier detection is increasingly urgent [2]. This paper also compares and analyze the efficiency of the proposed VAE model to the baseline methods selected as above.

Clustering algorithms based on DTW distance and Hausdorff distance generally divide N trajectories into several clusters through DBSCAN clustering algorithm. Finally, clusters with less than the threshold T_c are detected as outliers. In the clustering stage, the distance between trajectories needs

to be calculated and the clustering process needs to be carried out. Therefore, the theoretical time complexity of this kind of methods is $O(N^2)$, and it can reach $O(NlogN)$ after optimization. In the case of real-time detection, the distance between a given trajectory and all trajectories in the set needs to be calculated in the detection stage of these methods. Without considering the trajectory segmentation, the time consumption will increase linearly with the increase of the amount of trajectories in the set. For KNN and deep clustering methods, it is also necessary to calculate the distance in the clustering process, so its efficiency should be equal to or lower than DTW and Hausdorff.

For machine learning based methods, the training time of the model depends on the super parameters of the model, training set size and the convergence rate. Once the training is completed, only the distance between the trajectory itself and the reconstructed trajectory needs to be calculated in the detection phase, so the time complexity of the detection phase is $O(1)$. Because the VAE model proposed in this paper can be trained by sampling data set, the training data set is generally small and the training time is relatively short. While models SAE and GM-VSAE require training of all samples, so the training time is relatively long.

Under the same experiment conditions as in the previous section, set epochs = 200 and set the data set size N = [1000, 2000, 4000], we compare the clustering or training time and detection time of the baseline methods under different data sizes. For the clustering-based method, the training time refers to the calculation of the distance between trajectories and the clustering time; for the VAE method, the training time refers to the time required for the model training to converge. Detection time refers to the time taken to detect a single trajectory after clustering or model training is completed.

In the experiment, the DTW distance and Hausdorff distance were calculated by 32 threads in parallel. The experimental results are shown in Table 2. The training time and detection time in the table are the average of the experimental results.

Table 2. Computation efficiency of different methods (unit: second).

Methods	N = 1000		N = 2000		N = 4000		N > 100,000	
	Training	Detection	Training	Detection	Training	Detection	Training	Detection
DTW	31.65	0.35	135.376	1.304	513.228	5.446	-	-
Hausdorff	29.9053	1.62	126.713	10.696	514.683	61.490	-	-
SAE [32]	123.22	0.01	144.33	0.01	245.69	0.01	> 1000	<= 0.01
GM-VSAE [33]	124.39	0.01	146.82	0.01	268.24	0.01	> 1000	<= 0.01
VAE (ours)	**110.17**	**0.01**	**130.32**	**0.01**	**180.38**	**0.01**	**< 1000**	**<= 0.01**

It can be seen from the experimental results in Table 2 that when n = 1000, the VAE method consumes more training time than DTW and Hausdorff; When n = 2000, the VAE method takes about the same time as DTW and Hausdorff method; When n = 4000, the time-consuming of VAE method is significantly less than that of DTW and Hausdorff methods. This shows that the VAE method proposed in this paper has good convergence in the model training stage. The training time increases slowly with the growth of the data set size, so it can be applied to large-scale data set. In detection time, the VAE, SAE and GM-VSAE methods are all far less time-consuming than DTW and Hausdorff method, which can be ignored in practical application, because deep learning based method only needs to calculate the distance between the trajectory itself and the reconstructed trajectory. This again shows that the VAE based method proposed in this paper is very suitable for real-time detection of large-scale data set.

6. Conclusions

This paper introduces a trajectory outlier detection model based on variational auto-encoder in detail. Based on the statistical characteristics of the normal urban traffic trajectory data, the model converts the trajectory data into a distribution function using a variational auto-encoder, and optimizes the distribution parameters through historical data training to ensure that the generation probability of the normal original trajectory data is maximized when decoding. Finally, the outlier is detected by calculating the difference between the generated trajectory by the trained model and the original trajectory. The biggest advantage of the model proposed in this paper is that the outlier detection only needs to calculate the difference between the original trajectory and the generated trajectory by the model. Compared with the density-based methods and classification-based methods, the calculation amount is greatly reduced, which makes it very suitable for the real-time detection in large-scale data environments. In addition, the model can use the ratio of outliers in the verification data set to define the detection threshold of outliers, which can eliminate the difficulty of setting the distance threshold artificially, and make the applicable scenarios of the model more abundant. The experimental results on the real urban traffic trajectory data set show that the model proposed in this paper is very suitable for large-scale data real-time detection scenarios. In terms of effect, the precision and recall of the proposed model are over 95%, which is better than the methods we selected for comparison; in terms of efficiency, the model has good convergence in the training stage. The training time of the model increases slowly with the size of the data. The time consumption in the detection stage is a constant level, which is far better than some reference methods.

This paper verifies the effectiveness and efficiency of a trajectory outlier detection model based on variational auto-encoder. The data set used in the model training is urban traffic all-weather trajectory data. In view of the strong correlation between urban traffic trajectories and space-time, in practical applications, the data set can be spatiotemporal divided according to specific application needs, and model training can be carried out according to different spatiotemporal data set to obtain trajectory distribution characteristics under specific space-time, which can further improve the effect and efficiency of the model. This issue is worth further research in the future.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (No. 61801373) and Silk Road Research (2019YA07).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. Djenouri, D. Djenouri, J. Lin, Trajectory outlier detection: New problems and solutions for smart cities, *ACM Trans. Knowl. Discov. D*, **15** (2021), 1–28. https://doi.org/10.1145/3425867
2. Y. Djenouri, A. Belhadi, J. Lin, D. Djenouri, A. Cano, A survey on urban traffic anomalies detection algorithms, *IEEE Access*, **7** (2019), 12192–12205. https://doi.org/10.1109/ACCESS.2019.2893124
3. U. Ahmed, G. Srivastava, Y. Djenouri, J. C. W. Lin, Deviation point curriculum learning for trajectory outlier detection in cooperative intelligent transport systems, *IEEE Trans. Intell. Transp.*, **23** (2022), 16514–16523. https://doi.org/10.1109/TITS.2021.3131793
4. F. Meng, G. Yuan, S. Lv, Z. Wang, S. Xia, An overview on trajectory outlier detection, *Artif. Intell. Rev.*, **52** (2019), 2437–2456. https://doi.org/10.1007/s10462-018-9619-1
5. A. Belhadi, Y. Djenouri, C. Lin, A. Cano, Trajectory outlier detection: Algorithms, taxonomies, evaluation and open challenges, *ACM Trans. Manage. Inf.*, **11** (2020), 1–29. https://doi.org/10.1145/3399631
6. S. Wang, J. Cao, P. Yu, Deep learning for Spatio-Temporal data mining: A survey, *IEEE Trans. Knowl. Data Eng.*, **34** (2022), 3681–3700. https://doi.org/10.1109/TKDE.2020.3025580
7. Z. Liu, D. Pi, J. Jiang, Density-based trajectory outlier detection algorithm, *J. Syst. Eng. Electron.*, **24** (2013), 335–340. https://doi.org/10.1109/JSEE.2013.00042
8. J. Tang, H. Ngan, Traffic outlier detection by density-based bounded local outlier factors, *Inf. Technol. Ind.*, **4** (2016), 6–18. https://doi.org/10.17762/itii.v4i1.38
9. J. Mao, T. Wang, C. Jin, A. Zhou, Feature grouping-based outlier detection upon streaming trajectories, *IEEE Trans. Knowl. Data Eng.*, **29** (2017), 2696–2709. https://doi.org/10.1109/TKDE.2017.2744619
10. C. Piciarelli, G. L.Foresti, Anomalous trajectory detection using support vector machines, in *IEEE Conference on Advanced Video and Signal Based Surveillance*, (2007), 153–158. https://doi.org/10.1109/AVSS.2007.4425302
11. P. R. Lei, A framework for anomaly detection in maritime trajectory behavior, *Knowl. Inf. Syst.*, **47** (2016), 189–214. https://doi.org/10.1007/s10115-015-0845-4
12. J. Wang, Y. Yuan, T. Ni, Y. Ma, M. Liu, G. Xu, et al., Anomalous trajectory detection and classification based on difference and intersection set distance, *IEEE Trans. Veh. Technol.*, **69** (2020), 2487–2500. https://doi.org/10.1109/TVT.2020.2967865
13. L. X. Pang, S. Chawla, W. Liu, Y. Zheng, On mining anomalous patterns in road traffic streams, *Adv. Data Min. Appl.*, **2011** (2011), 237–251. https://doi.org/10.1007/978-3-642-25856-5_18
14. L. X. Pang, S. Chawla, W. Liu, Y. Zheng, On detection of emerging anomalous traffic patterns using GPS data, *Data Knowl. Eng.*, **97** (2013), 357–373. https://doi.org/10.1016/j.datak.2013.05.002
15. Q. Yu, Y. Luo, C. Chen, X. Wang, Trajectory outlier detection approach based on common slices sub-sequence, *Appl. Intell.*, **48** (2018), 2661–2680. https://doi.org/10.1007/s10489-017-1104-z
16. A. Belhadi, Y. Djenouri, D. Djenouri, T. Michalak, J. Lin, Deep learning versus traditional solutions for group trajectory outliers, *IEEE Trans. Cybern.*, **52** (2022), 4508–4519. https://doi.org/10.1109/TCYB.2020.3029338
17. Z. Zhang, G. Ni, Y. Xu, Comparison of trajectory clustering methods based on K-means and DBSCAN, in *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, **1** (2020), 557–561. https://doi.org/10.1109/ICIBA50161.2020.9277214

18. W. Dai, C. Zhang, X. Su, S. Cao, Trajectory outlier detection based on DBSCAN and velocity entropy, in *iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData and IEEE Cybermatics*, (2020), 550–557. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00097
19. C. Lyu, X. Wu, Y. Liu, Z. Liu, A Partial-Fréchet-Distance-Based framework for bus route identification, *IEEE Trans. Intell. Transp.*, **23** (2021), 9275–9280. https://10.1109/TITS.2021.3069630
20. B. K. Yi, H. V. Jagadish, C. Faloutsos, Efficient retrieval of similar time sequences under time warping, in *IEEE ICDE*, (1998), 201−208. https://doi.org/10.1109/ICDE.1998.655778
21. M. Vlachos, G. Kollios, D. Gunopulos, Discovering similar multidimensional trajectories, in *IEEE ICDE*, (2002), 673−684. https://doi.org/10.1109/ICDE.2002.994784
22. P. H. Daniel, K. Klara, M. K. Jon, On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane, in *Proceedings of the Eighth Annual Symposium on Computational Geometry*, (1992), 110−119. https://doi.org/10.1145/142675.142700
23. D. Zhang, Z. Chang, S. Wu, Y. Yuan, K. L. Tan, G. Chen, Continuous trajectory similarity search for online outlier detection, *IEEE Trans. Knowl. Data Eng.*, **34** (2022), 4690–4704. https://doi.org/10.1109/TKDE.2020.3046670
24. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444. https://doi.org/10.1038/nature14539
25. T. Fernando, S. Denman, S. Sridharan, C. Fookes, Soft + Hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection, *Neural Networks*, **108** (2018), 466–478. https://doi.org/10.1016/j.neunet.2018.09.002
26. A. Belhadi, Y. Djenouri, G. Srivastava, A. Cano, J. Lin, Hybrid group anomaly detection for sequence data: Application to trajectory data analytics, *IEEE Trans. Intell. Transp.*, **23** (2022), 9346–9357. https://doi.org/10.1109/TITS.2021.3114064
27. G. E. Hinton, Reducing the dimensionality of data with neural networks, *Science*, **313** (2006), 504–507. https://doi.org/10.1126/science.1127647
28. D. P. Kingma, M. Welling, Auto-encoding variational bayes, preprint, arXiv:1312.6114.
29. X. Chen, J. Xu, R. Zhou, W. Chen, J. Fang, C. Liu, TrajVAE: A Variational AutoEncoder model for trajectory generation, *Neurocomputing*, **428** (2021), 332–339. https://doi.org/10.1016/j.neucom.2020.03.120
30. J. Chen, S. Sathe, C. Aggarwal, D. Turaga, Outlier detection with autoencoder ensembles, in *Proceedings of the 2017 SIAM International Conference on Data Mining*, (2017), 90–98. https://doi.org/10.1137/1.9781611974973.11
31. D. Hendrycks, M. Mazeika, T. Dietterich, Deep anomaly detection with outlier exposure, preprint, arXiv:1812.04606.
32. P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, preprint, arXiv:1607.00148.
33. Y. Liu, K. Zhao, G. Cong, Z. Bao, Online anomalous trajectory detection with deep generative sequence modeling, in *IEEE ICDE*, (2020), 949–960. https://doi.org/10.1109/ICDE48307.2020.00087

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)