Unsupervised learning trajectory anomaly detection algorithm based on deep representation

Zhongqiu Wang1, Guan Yuan1,2,3, Haoran Pei1, Yanmei Zhang1 and Xiao Liu1

Abstract

Without ground-truth data, trajectory anomaly detection is a hard work and the result lacks of interpretability. Moreover, in most current methods, trajectories are represented by geometric features or their low-dimensional linear combination, and some hidden features and high-dimensional combined features cannot be found efficiently. Meanwhile, traditional methods still cannot get rid of the limitation of space attributes. Therefore, a novel trajectory anomaly detection algorithm is present in this article. Unsupervised learning mechanism is used to overcome nonground-truth problem and deep representation method is used to represent trajectories in a comprehensive way. First, each trajectory is partitioned into segments according to its open angles, then the shallow features at each point of a segment are extracted and. In this way, each segment is represented as a feature sequence. Second, shallow features are integrated into auto-encoder-based deep feature fusion model, and the fusion feature sequences can be extracted. Third, these fused feature sequences are grouped into different clusters using a unsupervised clustering algorithm, and then segments which quite differ from others are detected as anomalies. Finally, comprehensive experiments are conducted on both synthetic and real data sets, which demonstrate the efficiency of our work.

Keywords

Unsupervised learning, outlier detection, deep representation, trajectory data

Date received: 17 March 2020; accepted: 16 October 2020

Handling Editor: Yanjiao Chen

Introduction

With the continuous development and application of positioning equipment, such as global position system (GPS), Wi-Fi as well as wireless sensor network (WSN), a large volume of moving objects can be tracked with their trajectory data stored in the database. It has been a hot topic of analyzing and extracting knowledge from these data. Hawkins1 gave the definition of outlier as follows: an observation deviates greatly from others and arouses suspicion that it was generated by a different mechanism. Trajectory anomaly detection2,3 is an important research branch in the field of trajectory data mining. And its task is to discover trajectories or their segments which differ substantially from or are inconsistent with the remaining set. As for trajectory anomaly, an outlier means a data object that is grossly different from or inconsistent

1School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China
2Digitization of Mine, Engineering Research Center of Ministry of Education, Xuzhou, China
3Jiangsu Key Laboratory of Mine Mechanical and Electrical Equipment, China University of Mining and Technology, Xuzhou, China

Corresponding author:
Guan Yuan, School of Computer Science and Technology, China University of Mining and Technology, No. 1, Daxue Road, Xuzhou 221116, China.
Email: yuanguan@cumt.edu.cn
Traditional anomaly detection algorithms mainly consist of distance-based algorithms\(^{5,7}\) and density-based algorithms\(^{8,9}\) but unsupervised methods are rarely used. The distance-based algorithm mostly judges whether the trajectory is anomaly or not according to the space distance of the trajectory. Due to limitations in space distance comparison, the trajectory with the same characteristics but far distance may be detected as an exception, which will affect the accuracy of the trajectory anomaly detection. The density-based algorithm is essentially the extension of distance algorithm, which cannot fundamentally eliminate the limitations of distance-based trajectory anomaly detection algorithm.

In the traditional methods of trajectory anomaly detection, the trajectory is usually represented by hand extracted geometric features or low-dimensional linear combination of geometric features. These representation methods cannot find some hidden features or high-dimensional composite features. For example, Figure 1 shows the shape, angle change, distance change, and speed change of trajectory, respectively. By analyzing the internal characteristic attributes of trajectory such as angle, speed, and distance, it is found that the change degree is different. It can be seen that only measuring the linear combination of a certain geometric feature or a simple geometric feature is not enough to comprehensively reflect the movement trend of the trajectory. It is difficult to comprehensively describe the movement mode of the moving object’s trajectory. Therefore, error may exist in the result of anomaly detection because of incomplete feature extraction and incomplete representation of trajectory feature sequence.

In view of the shortcomings of existing research methods, this article proposes an unsupervised learning trajectory anomaly detection algorithm based on deep representation (TAD-FM). Through the transformation and processing of GPS data, the trajectory is described comprehensively with deep characteristics. Compared with algorithms based on distance and density, the deep feature fusion model based on auto-encoder is proposed in this article, which transforms the geometric feature representation of trajectory into sequence pattern, represents trajectory by feature sequence, and desalinates the spatial attribute of trajectory itself. Therefore, the limitation of using spatial distance to judge anomalies is fundamentally solved. At the same time, from the perspective of fully exploring the characteristics of the trajectory, the low-dimensional shallow features and the high-dimensional deep features of the trajectory are integrated to describe the trajectory comprehensively. So as to characterize the overall trajectory features according to the description sequence to detect the trajectory anomaly.

The main framework of this article is given in Figure 2 with parts of ideas borrowing from Chen et al.\(^{10}\) GPS trajectory data can be obtained from a variety of positioning devices. The original GPS trajectory data are a three-dimensional (3D) data composed of longitude, latitude, and time. In the trajectory data division stage, the trajectory is divided into segments according to the open angle. In the stage of deep feature extraction, the trajectory segment is transformed into shallow feature sequence first, and then the shallow feature sequence is put into the deep feature fusion model based on auto-encoder to obtain the deep feature sequence representation of trajectory segment, that is, the original GPS data of trajectory segment are transformed into the fusion feature sequence. In the clustering stage, in order to obtain the trajectory segments of different motion modes, the density-based spatial clustering of applications with noise (DBSCAN) algorithm is used to cluster by measuring the cosine distance between fusion feature sequences. In the phase of anomaly detection, the anomaly trajectory segments are extracted by comparing the cosine similarity between the fusion feature sequences, and then consider the proportion of the anomaly trajectory segments in the overall trajectory, finally detect the anomaly trajectory.

The main contributions of this article are as follows:

1. On the basis of the idea of deep embedding, a deep feature fusion model based on auto-encoder is proposed to reduce human
intervention, extract high-dimensional combined features of trajectory, transform the geometric feature sequence of trajectory into the mixed feature sequence of high-dimensional non-linear representation, and describe the trajectory comprehensively.

2. The geometric representation of the trajectory is transformed into a sequential pattern. The limitation of using spatial distance to judge the anomaly is fundamentally solved using the way of feature sequence to represent the trajectory and desalinate the spatial attribute of the trajectory itself.

3. Experiments based on multiple data sets show that the algorithm proposed in this article can discover the anomaly trajectory, which proves the effectiveness and feasibility of the algorithm.

The organization structure of this article is as follows. In section “Related work,” the related work is described. In section “Trajectory representation,” the method of trajectory division is introduced. In section “Feature extraction and transformation,” the deep feature fusion model based on auto-encoder is proposed to describe how to extract and transform the features from the divided trajectory segments. In section “Unsupervised learning clustering,” the multi-category clustering method based on unsupervised learning is introduced. In section “Trajectory outlier detection,” the method of trajectory anomaly detection is described.

Figure 2. The framework of this article.
in detail. In section “Experimental analysis,” the algorithm is evaluated by experiments. In section “Some real application scenarios,” some typical application scenarios are discussed. In section “Conclusion,” the full text is summarized, and the future research direction is pointed out.

**Related work**

**Trajectory anomaly detection**

**Distance-based algorithm.** Knorr et al.\(^7\) not only regarded the trajectory as a series of points but also transformed the trajectory into a feature sequence of position, direction, and speed. By comparing the distance, the trajectory anomaly was detected. However, this method only compares the overall characteristics of the trajectory while ignores the local characteristics. As the trajectory grows, the problems exposed by this method become more and more obvious. In order to solve the problem caused by comparing the whole trajectory, Lee et al.\(^5\) proposed the density-based trajectory outlier detection (TRAOD) algorithm. The TRAOD algorithm consists of two steps: division and detection. In the division step, the trajectory is divided into several segments, which are represented by the line segments connected by two sampling points. In the detection step, the distance between the line segments is used to measure its matching, so as to detect the trajectory anomaly. This method solves the problem of complex trajectory anomaly detection, but only by describing the trajectory segment as a directed segment composed of the start and end points, the characteristics of the trajectory itself are inevitably ignored. Yuan et al.\(^6\) put forward the trajectory structure framework and introduced it into the trajectory anomaly detection. By considering the internal characteristics of the trajectory, the accuracy of the analysis is improved, so as to avoid the problem of only comparing the spatial shape of the trajectory. However, due to the different importance of each internal feature of the trajectory, subjective weight will interfere with the results, thereby affecting the accuracy of anomaly detection. Most distance-based algorithms judge whether the trajectory is anomaly or not according to the space distance of the trajectory. Because of the limitation of space distance comparison, it is inevitable that the trajectory with the same characteristic but a long distance is detected as an exception, which affects the accuracy of the trajectory anomaly detection.

**Density-based algorithm.** Breunig et al.\(^8\) used local anomaly factor (LOF) to determine whether the trajectory is anomaly, but a large number of neighbor queries are required to calculate LOF, so the cost of calculation is very high.\(^11\) Liu et al.\(^9\) introduced the concept of trajectory density and proposed density-based trajectory outlier detection (DBTOD) algorithm to make up for the shortcomings that TRAOD algorithm cannot detect local and dense trajectories. The density between trajectories is calculated by distance, so the density-based algorithm is essentially an extension of distance-based algorithm, which cannot fundamentally solve the limitations of the distance-based algorithm.

**Embedding**

Sutskever et al.\(^12\) first proposed the method of sequence-to-sequence auto-encoder for machine translation. Chen et al.\(^10\) used auto-encoder fusing a variety of different features to get more semantical and discriminative context representation in the latent space, so as to analyze the taxi destination. Liu et al.\(^13\) used embedding method for real-time personalized search and similar product list recommendation. Dong et al.\(^14\) proposed a novel Auto-encoder Regularized deep Neural Network (ARNet) and a trip encoding framework trip2vec to learn drivers’ driving styles directly from GPS records, by combining supervised and unsupervised feature learning in a unified architecture. Embedding method is widely used. Through the analysis of the above methods, this article applies the embedding method to trajectory anomaly detection. Beritelli et al.\(^15\) proposed a radial basis probabilistic neural network to classify non-linear and unstable data. The implemented method has been devised in order to slowly increase, during training, the generalization capabilities of a radial basis probabilistic neural network classifier as well as preventing it from over-generalization and the consequent lack of resulting classification performances.

**Trajectory representation**

The temporal distance between two consecutive points in the same trajectory is relatively small (with respect to the mobility phenomenon under consideration). In this view, the trajectory is simply a sequence of relatively close in time ordered points.\(^16\) Generally speaking, the segmentation task splits a sequence of data points, in a series of disjoint subsequences consisting of points that are homogeneous with respect to some criteria.\(^17,18\) Therefore, this article uses the divided trajectory segment as the basic unit of analysis.

Maike et al.\(^19\) defined the segmentation problem as follows: to split a high-sampling rate trajectory into a minimum number of segments such that the movement characteristic inside each segment is uniform in some sense. The purpose of dividing a trajectory into segments is that trajectory segments contain part of the original trajectory data and hidden information, so they can be more flexible to combine and describe
The key of trajectory division is to find some key points that can divide the trajectory into multiple smooth trajectory segments. Through the key points to divide the trajectory, it can effectively divide the trajectory segments that retain trajectory characteristics and are relatively smooth. If the trajectory is not divided and the whole trajectory is taken as the input data, the analysis of trajectory data is limited to the original trajectory data, and these are also a kind of time series data. Empirical studies showed that simply treating raw trajectory data as 3D signal inputs for either traditional machine learning or deep learning algorithms just does not work. Therefore, in order to make the trajectory data suitable for the deep neural network algorithm, it is necessary to extract the trajectory features with stronger semantic expression from the original trajectory data. The original trajectory data are represented by feature sequence.

Figure 3. The open angle of trajectory.

In order to describe the process of track division more clearly, a simple example is used to illustrate the trajectory division. Figure 4 shows the schematic diagram of trajectory segment division. According to Definition 2, \( \alpha_1 \) is calculated as 119°, \( \alpha_2 \) as 128°, and the threshold \( \omega \) is set as 125°. It can be seen that \( \alpha_1 \) is less than the threshold value, \( \alpha_2 \) is greater than the threshold value, then \( \alpha_2 \) is the open angle, \( P_1 \) is the key point, and the trajectory is divided into two trajectory segments at \( P_1 \).

**Feature extraction and transformation**

Deep neural networks have been proved powerful in learning from speech data,23,24 text data,25 and image data.26 Trajectory data are similar to language and text data, and these are also a kind of time series data. However, according to Definition 1, trajectory data are 3D data composed of two-dimensional (2D) coordinates and timestamps. Empirical studies showed that simply treating raw trajectory data as 3D signal inputs for either traditional machine learning or deep learning algorithms just does not work. Therefore, in order to make the trajectory data suitable for the deep neural network algorithm, it is necessary to extract the trajectory features with stronger semantic expression from the original trajectory data. The original trajectory data are represented by feature sequence.

As shown in Figure 5, the core of feature extraction and transformation is divided into two steps. First, the geometric features are extracted from several indefinite length original trajectory segments and transformed into fixed length shallow feature sequence \( F \) through dimension expansion. Second, the deep feature fusion model based on auto-encoder is used to transform shallow feature sequence into mixed feature sequence \( FM \), that is, shallow feature sequence \( F \) is spliced with deep trajectory feature sequence \( FD \) to get mixed feature sequence \( FM \).

**Shallow feature sequence extraction**

The trajectory segment \( TS \) contains rich information, from which a variety of information can be extracted to represent the movement trend and structural features of the trajectory segment. In this article, four representative geometric features are extracted from the trajectory segment: velocity change \( f_v \), distance change \( f_d \), angle change \( f_a \), and acceleration change \( f_{acc} \). However, GPS data may have errors in recording, and the influence of different degrees of change in the same feature on trajectory segments is different. In order to reduce the inaccuracy of feature extraction caused by possible data errors, so that the relevant features can be represented in a more detailed way, the original geometric features need to be expanded.
Definition 3. Shallow feature sequence: A frame encoding statistics of the basic features is called the shallow feature of the trajectory. The set \( Q \) composed of \( n \) shallow trajectory feature sequences is called shallow feature sequence set, which is expressed as \( Q = \{ F_1, F_2, F_3, ..., F_n \} \). Each shallow feature sequence \( F_i \) represents a trajectory segment \( T S_i \), \( 1 \leq i \leq n \).

Dimension expansion requires calculating mean, minimum, maximum, 25%, 50%, and 75% quartiles, and standard deviation of each trajectory segment feature. The calculated results are the shallow feature sequences of each trajectory segment. According to Definition 3, this article analyzes the velocity variation \( f_v \), distance variation \( f_d \), angle variation \( f_a \), and acceleration variation \( f_{acc} \) of the trajectory. The shallow feature sequence \( F \) is expressed as \( F = \{f_v, f_d, f_a, f_{acc}\} \times \{\text{mean}, \text{minimum}, \text{maximum}, 25\% \text{ quartiles,} 50\% \text{ quartiles,} 75\% \text{ quartiles, standard deviation}\} \). Obviously, in this article, the shallow feature sequence \( F \) is a 28(4 \times 7) dimensional feature sequence. The shallow feature sequence can reflect the trajectory segment intuitively and quantify the original GPS trajectory data at the same time. The trajectory segment set is represented by the shallow feature sequence set \( Q \), which is convenient for later analysis and processing.

Deep feature fusion

A deep feature fusion model based on auto-encoder is proposed. The shallow feature sequence is transformed into deep feature sequence using the model, and the shallow feature sequence and deep feature sequence are spliced into mixed feature sequence, so that the features of trajectory segments can be represented comprehensively from deep and shallow aspects.

Auto-encoder. By analyzing the trajectory segments represented by the intuitionistic feature sequences, we can find the anomalies. However, the shallow feature sequence can reflect the trajectory segment intuitively and quantify the original GPS trajectory data at the same time. The trajectory segment set is represented by the shallow feature sequence set \( Q \), which is convenient for later analysis and processing.
Figure 6. The process of auto-encoder.

sequence is only the feature representation of low-dimensional trajectory segments, which will limit the performance of machine learning algorithm. The typical problem is that the potential feature information and the association between different types of combined features cannot be found. Therefore, this article uses auto-encoder to solve this problem. Auto-encoder belongs to the embedding method, which is an unsupervised artificial neural network. After many training, it can capture the most significant features in the input data.

Figure 6 shows a three-layer auto-encoder model. $L_1$ is the input layer, $L_2$ is the hidden layer, and $L_3$ is the output layer. Among them, the input layer $L_1$ and the hidden layer $L_2$ constitute the encoder, which are responsible for extracting potential features. The hidden layer $L_2$ and the output layer $L_3$ constitute the decoder, which are responsible for reconstructing the input data from the hidden layer. The data in input layer $L_1$, hidden layer $L_2$, and output layer $L_3$ are represented by $x$, $h$, and $y$ ($x, y \in [0, 1]^n$, $h \in [0, 1]^m$). $W$ is the weight matrix ($W \in R^{m \times n}$), $b_1$ is the offset of input layer $L_1$ ($b_1 \in R^m$), and $b_2$ is the offset of hidden layer $L_2$ ($b_2 \in R^n$).

The value $h$ in hidden layer $L_2$ and the value $y$ in output layer $L_3$ are calculated by equations (2) and (3), where $\sigma(\cdot)$ represents sigmoid activation function

$$h = f(x) = \sigma(Wx + b_1)$$

$$y = g(h) = \sigma(W^zh + b_2)$$

The purpose of using auto-encoder is to extract the significant high semantic features of the original data from the hidden layer; at the same time, it has the function of reducing dimension and noise. In order to make the learned hidden layer meaningful, the data output by the encoder needs to enter the decoder again for decoding. Through repeated iterations, the input and output parameters are updated to make the value of $y$ and $x$ close, that is, the target optimization function of $y = F_w, \theta(x) = x$ is learned from the encoder. The target optimization function can be represented by the reconstruction error function $L(x, y)$, as shown in equation (4). According to the reconstruction error function, loss function can be obtained, which is calculated by equation (5).

$$L(x, y) = - \sum_{i=1}^{n} (x_i \log(y_i) + (1 - x_i) \log(1 - y_i))$$

$$Loss = \sum_{x \in Q} L(x, g(f(x)))$$

The updating methods of weight matrix $W$ and offset quantities $b_1$ and $b_2$ are shown in equations (6)–(8).

$$W_{new} = W_{old} - \alpha \frac{\partial loss}{\partial W_{old}}$$

$$b_{1 new} = b_{1 old} - \alpha \frac{\partial loss}{\partial b_{1 old}}$$

$$b_{2 new} = b_{2 old} - \alpha \frac{\partial loss}{\partial b_{2 old}}$$

In this article, in order to make the results more practical, only the vectors in the hidden layer $L_2$ represented by the high-dimensional feature sequence after feature compression, that is, after feature reconstruction, are selected as the output results of the auto-encoder.

Mixed feature sequence transformation. In this section, the deep feature fusion model based on auto-encoder is introduced in detail. In order to express conveniently, the definition of deep feature sequence and mixed feature sequence is introduced first.

Definition 4. Deep trajectory feature sequence: The deep trajectory feature sequence $FD$ is obtained by depth representation of the shallow trajectory feature sequence $F$. The set $QD$ composed of $n$ deep trajectory feature sequences is called deep feature sequence set, which is expressed as $QD = \{FD_1, FD_2, FD_3, \ldots, FD_n\}$. Among them, each deep feature sequence $FD_i$ represents a trajectory segment $TS_i$ and is also the deep representation of the shallow feature sequence $F_i$.

Definition 5. Mixed feature sequence: The feature sequence $F_i$ and the deep feature sequence $FD_i$, which represent the same trajectory segment $TS_i$, are spliced, and the spliced sequence is expressed in high-
dimensional depth. The result sequence is the mixed feature sequence $FM_i$. The set of all mixed feature sequences is called mixed feature sequence set $MD$, which is expressed as $MD = \{FM_1, FM_2, FM_3, ..., FM_n\}, 1 \leq i \leq n$.

The shallow feature sequence and the deep feature sequence can express the shallow and deep features, respectively. Using the deep feature fusion model based on auto-encoder, the shallow feature sequence and the deep feature sequence are transformed into the mixed feature sequence, which not only considers the detail information of the shallow feature but also combines the semantic information of the deep feature. So, the shallow and deep features can be highlighted at the same time.

Figure 7 shows the deep feature fusion model based on auto-encoder. By inputting the shallow feature sequence into auto-encoder 1, we can get the deep feature sequence shown in Definition 4. Using the deep feature sequence $FD$ to express the trajectory segment in a higher dimension, we can capture the relationship between the potential valuable feature information in the trajectory segment and a variety of non-linear combined features. Then, the deep feature sequence and the shallow feature sequence are spliced into the auto-encoder 2. The mixed feature sequence shown in Definition 5 can be obtained. The weight and offset of the feature sequence put into the auto-encoder are calculated by equations (6)–(8). Using the fusion feature sequence $FM$, the trajectory segments are described comprehensively from the two aspects of the shallow feature and the deep feature.

Unsupervised learning clustering

The purpose of unsupervised learning–based multi-category clustering is to divide different types of trajectory segments into different clusters. The trajectory segments that need to be detected should exist in the same neighborhood, that is, the same cluster with similar motion mode, rather than all the trajectory segments in the set. This is conducive to reducing the complexity of the algorithm and also has practical significance.

However, traditional clustering methods are often based on hand-crafted features. These features have limited capacity and cannot dynamically adjust to capture the prior distribution. Therefore, this article uses the features processed by the deep feature fusion model based on auto-encoder to cluster and breaks through the limitation of manual feature extraction for the overall representation of trajectory.

Definition 6. Cosine similarity: The method to evaluate the similarity of two mixed feature sequences by calculating the cosine value of the angle between them is called cosine similarity. Equation (9) shows the cosine similarity of the mixed feature sequence $FM_i$ and $FM_j$, $1 \leq i, j \leq n$

$$s_{ij} = \frac{FM_i \cdot FM_j}{FM_i, FM_j}$$

The purpose of using cosine similarity to measure distance is that most of the current trajectory clustering algorithms rely on the spatiotemporal similarity measurement of trajectory. They cannot cluster trajectories with similar characteristics that exist in different spatio-temporal regions. In this article, the mixed feature sequence is used to represent the trajectory segments, and all the trajectory segments are described as space vectors of common starting point. The cosine similarity is used to measure the distance between them. The vectors pointing to a certain range are grouped into one category, that is, the trajectory segments with similar
features are grouped into one cluster, which breaks through the limitation of space-time.

In this article, the density-based clustering algorithm DBSCAN\textsuperscript{30} is used to cluster, and the cosine similarity shown in Definition 6 is used to measure the distance. DBSCAN can find clusters of any shape and can find irregular clusters without knowing the number of clusters in advance. The unsupervised trajectory anomaly detection algorithm proposed in this article has no clear attribute label for the attribute type of trajectory, and the subjective specified clustering number will affect the classification of trajectory. Therefore, the DBSCAN algorithm has a high applicability.

**Definition 7.** \(\varepsilon\)-neighborhood: If the mixed feature sequence \(FM_i\) and \(FM_j\) are in MD and the cosine similarity does not exceed the nearest neighbor threshold \(\varepsilon\), then the \(\varepsilon\) nearest neighbor of the mixed feature sequence \(FM_i\) is expressed as \(N_\varepsilon(FM_i)\), which is expressed by equation (10), where \(s_{ij}(FM_i, FM_j)\) indicates the cosine similarity between \(FM_i\) and \(FM_j\)

\[
N_\varepsilon(FM_i) = \{FM_j \in MD \cap i \neq j \mid s_{ij}(FM_i, FM_j) \leq \varepsilon\}
\] (10)

**Definition 8.** Core sequence: For any mixed feature sequence \(FM_i \in MD\), if \(|N_\varepsilon(FM_i)| \geq MinPts\), \(FM_i\) is the core sequence.

**Definition 9.** Directly density-reachable: If \(FM_i \in N_\varepsilon(FM_j)\) and \(|N_\varepsilon(FM_j)| \geq MinPts\), \(FM_i\) is said to be directly density-reachable by \(FM_j\).

**Definition 10.** Density-reachable: If there is a series of mixed feature sequences \(FM_i, FM_{i+1}, ..., FM_r, FM_{j-1}\), where \(FM_p\) and \(FM_{p+1}\) are directly density-reachable, then \(FM_i\) is said to be density-reachable by \(FM_j\). Density-reachable does not have symmetry.

**Definition 11.** Cluster: If a non-empty set \(C\) satisfies the following two conditions, then it is called cluster

1. Connectivity: For \(\forall FM_i, FM_j \in C\), \(FM_i\) and \(FM_j\) are density connected.
2. Maximal: for \(\forall FM_i, FM_j\), if \(FM_i \in C\) and \(FM_j\) can be reached by \(FM_i\) density, then there is \(FM_j \in C\).

**Definition 12.** Noise: In the cluster set \(Cl = \{C_1, C_2, C_3, ..., C_{m\{m\}}\}\), the mixed feature sequence \(FM_k\) that exists in the mixed feature sequence set MD but does not belong to any clustering \(C_i\), is called noise, that is, noise = \(\{FM_k \in MD \mid FM_k \notin C_i\}\).

For the problem of noise treatment, one view holds that the noise is composed of data points that are not suitable for clustering structure, so it can be considered as divergence.\textsuperscript{31} A simple processing solution is introducing a tolerance on the number of noise points inside a segment to smooth out the noise.\textsuperscript{32} But another point of view is that noise contains important semantic information and cannot be ignored in dynamic phenomena. In some areas, specialists use the term “excursion” to characterize the temporary absence from the home-range where the animals reside.\textsuperscript{33}

In this article, GPS coordinates smoothed by Kalman filtering is taken into consideration.\textsuperscript{19} At the same time, through feature extraction and transformation, a single trajectory point can be mapped into multi-dimensional features, which can dilute and smooth the possible noise caused by sampling equipment to the maximum extent. Therefore, noise is considered as a part of real data and classified as anomaly.

As shown in Figure 8, the vectors \(a\)–\(h\) with different directions in the graph are the trajectory segments represented by the mixed feature sequence, respectively. According to the DBSCAN clustering algorithm, they are divided into three clusters: cluster 1, cluster 2, and cluster 3. In cluster 1, \(a\) is the core sequence, and \(b\) is in the \(\varepsilon\)-neighborhood of \(a\). So, \(b\) is directly density-reachable. At the same time, the direct density of \(b\) and \(c\) can be reached, so \(c\) is density-reachable. The setting of MinPts and \(\varepsilon\) will change the size of the sector and the number of sector inwards. It can be seen that the directions of vectors are different among different clusters, but the directions of vectors in the same cluster are roughly the same. It can be concluded that the trajectory segments in the same cluster have the same motion mode.

The specific algorithm pseudo code of clustering based on unsupervised learning is shown in Algorithm 1. In lines 04–10, the \(\varepsilon\)-neighborhood value of each unclassified mixed feature sequence is calculated to determine whether it meets the conditions of the core sequence. If it is the core sequence, the mixed feature sequence in its neighborhood is inserted into the cluster.
Algorithm 1: unsupervised learning clustering algorithm

Input: MD = \{FM_1, FM_2, ..., FM_m\}, \varepsilon, MinPts
Output: cluster set C = \{C_1, C_2, C_3, ..., C_m\}
(01) mark all mixed feature sequences in MD as unclassified;
(02) C = NULL; /* initialization C as empty */
(03) for each FM \in MD do
(04) if FM unclassified then
(05) if N_i(FM) < MinPts then
(06) mark FM as noise;
(07) else
(08) mark FM as core sequence;
(09) C = new cluster; /* create a new C*/
(10) insert sequence in FM neighborhood into C;
(11) for each FM \in N_i(FM) do
(12) if N_i(FM) > MinPts then
(13) insert sequence in FM neighborhood that is not included in other clusters into C;
(14) end for
(15) C \leftarrow C; /*insert C into C*/
(16) end for

In lines 11–14, the mixed feature sequence with the density of the core sequence can be calculated. The mixed feature sequence which is not classified in the neighborhood is scanned and its \(\varepsilon\)-neighborhood is calculated. If it is the core sequence, all the mixed feature sequences in its neighborhood are inserted into the cluster.

The time complexity of the algorithm is \(O(n^2)\). If the index tree is used, the data retrieval can be changed from the original global space retrieval to the local node retrieval, and the time complexity can be reduced to \(O(n \log n)\), where \(n\) represents the number of fusion feature sequences.

Trajectory outlier detection

In this section, the specific method of trajectory anomaly detection will be described in detail. In order to express clearly, some necessary definitions are given here.

Definition 13. Cosine similarity matrix: Calculate the cosine similarity of each element in the mixed feature sequence set MD, and the resulting symmetric matrix is the cosine similarity matrix \(S\), as shown in equation (11). Among them, \(s_{ij}\) represents cosine similarity of mixed feature sequence \(FM_i\) and \(FM_j\), \(1 \leq i, j \leq n\)

\[
S = \begin{bmatrix}
  s_{11} & s_{12} & \cdots & s_{1n} \\
  s_{21} & s_{22} & \cdots & s_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  s_{n1} & s_{n2} & \cdots & s_{nn}
\end{bmatrix}
\] (11)

The calculation method of each element in cosine similarity matrix \(S\) is shown in equation (3). The element \(s_{ij}\) in row \(i\) and column \(j\) represents the cosine similarity of mixed feature sequence \(FM_i\) and \(FM_j\), which is expressed as \(s_{ij} = \frac{FM_i \cdot FM_j}{\|FM_i\| \|FM_j\|}\).

Definition 14. Similarity index: The sum of row \(k(1 \leq k \leq n)\) in matrix \(S \sum s_{ik}\), which represents the similarity index \(ano_i\) of the mixed feature sequence \(FM_i\), that is, \(ano_i = \sum s_{ik}(1 \leq k \leq n)\). Because \(FM_i\) is the feature sequence representation of \(TS\)\(ano_i\) also represents the similarity index of the trajectory segment \(TS\). The set of similarity index of all trajectory segments is called similarity index set, expressed as \(A = \{ano_1, ano_2, ano_3, ..., ano_n\}\).

Definition 15. Anomaly trajectory segment: Given the threshold value \(\rho\) of similarity index, for a trajectory segment \(TS\), if its similarity index is less than the threshold value \(\rho\), then the trajectory segment \(TS\) is an anomaly trajectory segment.

Definition 16. Anomaly trajectory: Given the threshold value \(\tau\) of anomaly length, in a trajectory \(T = \{TS_1, TS_2, TS_3, ..., TS_n\}\), if the length ratio of anomaly trajectory segment \(TS\) is greater than the threshold \(\tau\), then this trajectory is called an anomaly trajectory.

After clustering, the cluster set \(C\) containing each cluster can be obtained. First, the cosine similarity matrix of each cluster was calculated according to Definition 13; second, the similarity index set was calculated according to Definition 14; then, according to definition 15, we can get the anomaly trajectory segment; finally, the anomaly trajectory is determined through Definition 16.

Taking Figure 9 as an example. It contains two clusters and multiple trajectory segments. The red lines represent the anomaly trajectory segments, and the black or green ones are the normal trajectory segments. Among them, the noise is the trajectory segment that does not belong to any clusters. According to the previous statement, it is classified as an anomaly first. \(TS_1\), \(TS_2\), and \(TS_3\) are trajectory segments after trajectory \(T\) division. After feature transformation and clustering, \(TS_1\) is grouped into cluster 1 and \(TS_2\) and \(TS_3\) are grouped into cluster 2. For the trajectory segment, because the similarity index \(ano_3\) of \(TS_3\) is less than the given threshold value \(\rho\) of similarity index, it is determined that \(TS_3\) is an anomaly trajectory segment. At the same time, for the whole trajectory, the length proportion of the anomaly trajectory segment \(TS_3\) in trajectory \(T\) is greater than the threshold value \(\tau\) of the anomaly length proportion, so trajectory \(T\) is an anomaly trajectory.
The specific algorithm pseudo code of trajectory anomaly detection is shown in Algorithm 2. It is easy to see that the whole algorithm can be divided into two parts: lines 02–10 calculate the cosine similarity matrix and similarity index set of each cluster in the cluster set. If the similarity index of a trajectory segment is less than the similarity index threshold \( r \), it is an anomaly trajectory segment. In lines 11–15, if the length proportion of the anomaly trajectory segment in the whole trajectory is greater than the threshold \( t \), it is an anomaly trajectory.

The analysis of the algorithm shows that the time complexity of the algorithm is \( O(m^3n) \), where \( m \) represents the number of elements in the trajectory set \( Cl \) and \( n \) represents the number of elements in the similarity index set \( A \).

**Experimental analysis**

In this section, abundant experiments are conducted to verify the effectiveness and feasibility of the proposed algorithms. Two data sets, synthetic trajectories\cite{34} and North Atlantic hurricanes\cite{35} are used in experiments. The operation environment required for the experiment is shown in Table 1.

**Data set**

**Synthetic trajectories.** Synthetic trajectory data set is a labeled data set proposed by Piciarelli et al.\cite{36} The data set includes 1000 subsets, each of which is composed of 260 randomly generated trajectories with the length of 16. The trajectories are divided into five clusters. Of the 260 trajectories, 250 are normal trajectories and the remaining 10 are anomaly trajectories. As shown in Figure 10(a), it is a subset of the data set, in which light (blue) represents normal trajectory and dark (red) represents anomaly trajectory.

**North Atlantic hurricanes.** North Atlantic hurricanes data set records the hurricane data of the North Atlantic from 1950 to 2006, with a total of 861 trajectories composed of 23,043 points. The information in the data set includes the longitude, latitude, wind speed, pressure, level, and sampling time of the hurricane center. The graphical representation of the data set is shown in Figure 10(b).

**Algorithm parameters analysis**

The parameters involved in trajectory anomaly detection algorithm include open angle threshold \( \omega \), learning rate, neighbor threshold \( \epsilon \), minimum number threshold \( MinPts \), anomaly index threshold \( \rho \), and anomaly segment length proportion threshold \( \tau \). The setting of these parameters directly affects the outcome of trajectory anomaly detection. The meaning of parameters in this algorithm is intuitive and clear. In different application fields, experts in the field can easily determine the value of parameters. In this subsection, we choose some subsets from synthetic trajectory data set to

![Figure 9. Example of anomaly detection.](image-url)
discuss how parameters affect the anomaly detection results.

Open angle threshold $\omega$. The open angle threshold $\omega$ directly determines the number of trajectory segments. If the $\omega$ setting is too excessive, the number of trajectory segments will be divided too much and the details of important features will be lost easily. If the $\omega$ setting is too low, the number of trajectory segments will be too small, and it will not be sensitive to mutation points, which will affect the clustering effect. Figure 11 shows the influence of different $\omega$ on the number of trajectory segments. In this experiment, five subsets represented as sub_data set 1–6 are selected randomly from synthetic trajectory data set. The curve in the figure represents the change in trajectory segment’s number divided by all trajectories in each data set under different open angle threshold $\omega$. It can be seen that when $\omega$ is 140, the curve gradually becomes flat.

Learning rate. The setting of learning rate will directly affect the optimization result of the loss function in the auto-encoder. Too excessive or too low learning rate cannot reduce the value of the loss function to the minimum, thus affecting the precision of the mixed feature sequence FM. When the learning rate is too excessive, loss function oscillation and gradient explosion may occur. When the learning rate is too low, there will be over fitting or slow convergence. Figure 12 shows the loss function diagrams corresponding to different learning rates. The six curves, $\text{loss} 1$–$\text{loss} 6$, marked in the figure correspond to the loss functions with learning rates of 0.05, 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. It can be clearly seen from the figure that the learning rate of loss function $\text{loss} 1$ and $\text{loss} 2$ is too low, so the convergence speed is very slow. The learning rate of loss function $\text{loss} 6$ is set too excessive, which leads to the phenomenon of oscillation, and the loss function cannot converge.

Nearest neighbor threshold $\epsilon$ and the minimum number of MinPts. The setting of nearest neighbor threshold $\epsilon$ and
minimum number of $\text{MinPts}$ directly affects trajectory segments clustering results. If $\text{MinPts}$ is set to a great value, most of trajectory segments will not meet the condition of $N_c(\text{TS}) \geq \text{MinPts}$. In most extreme cases, no trajectory segments meet such condition, which may result in all the trajectory segments being marked as noise. If $\text{MinPts}$ is too low, the number of core trajectory segments may increase and result in too many clusters. Meanwhile, if $\varepsilon$ is too excessive, trajectory segment that does not belong to any clusters may be clustered as a new one. While $\varepsilon$ is too low, there will be a large number of smaller clusters, which will also lead to a large number of trajectory segments cannot be in the neighborhood, thus becoming noise. Figure 13(a)–(d) shows the influence of $\text{MinPts}$ being too excessive, $\text{MinPts}$ being too low, $\varepsilon$ being too excessive, and $\varepsilon$ being too low on clustering, respectively. The data come from a subset of synthetic trajectory data set. In order to visualize cluster results, we use t-distributed stochastic neighbor embedding (t-SNE) algorithm\textsuperscript{37} to compress multi-dimensional data into two-dimensional points. The brown dots in the graph represent noise. As can be seen from Figure 13(a), when $\text{MinPts}$ is too excessive, a large amount of noise will be generated. Figure 13(b) and (d) shows that too small $\text{MinPts}$ or $\varepsilon$ can cause noise and too many clusters. Figure 13(c) shows that too excessive $\varepsilon$ can make two different clusters be wrongly divided into one cluster. Therefore, it is important to choose the nearest neighbor threshold $\varepsilon$ and the minimum number of $\text{MinPts}$.

**Similarity index threshold $\rho$.** In the process of anomaly detection, the similarity index obeys normal distribution. At the same time, it can avoid the deviation of detection result caused by the inaccurate threshold value $\rho$ of similarity index. The threshold value of similarity index is determined by $k\sigma$ criterion. It is more reasonable when $k$ is 1.645.\textsuperscript{38} Figure 14 shows the distribution of the similarity index $\text{ano}$. The abscissa represents the value of the similarity index $\text{ano}$ after Z-score standardization, and the ordinate represents the probability corresponding to each $\text{ano}$. The curve is generated after fitting the data.

In order to conform the actual situation, the threshold value $\rho$ only takes the boundary value on the left side, and the $y(\text{ano}_{i})$ as shown in equation (12) can be obtained, where 0 means anomaly and 1 means normal, that is, $\rho = \mu_{\text{ano}} - k\sigma_{\text{ano}}$

$$y(\text{ano}_{i}) = \begin{cases} 0, & \text{ano}_{i} \leq \mu_{\text{ano}} - k\sigma_{\text{ano}} \\ 1, & \text{else} \end{cases}$$

**Experimental results analysis**

**Synthetic trajectories.** In this article, labeled artificial trajectory data set is used first to verify the unsupervised trajectory anomaly detection algorithm based on deep representation; then, we compare the detected anomaly with the label in the data set. The validity and feasibility of the algorithm are proved by high accuracy.

Figure 15 shows the loss function, with orange (light) representing the mixed feature sequence $FM$ and blue (dark) representing the deep feature sequence $FD$. The learning rate sets to 0.5. After iteration, the loss of $FM$ is gradually stable at 0.0396 and that of $FD$ is gradually stable at 0.0389. The loss function is given by equation (5) in section “Auto-encoder.” It can be seen from the figure that the loss of $FM$ and $FD$ starts to decrease slowly when iterating 2000 times and 1000 times, respectively. Compared with $FM$, the
The convergence speed of FD is faster. This is because FM contains more features, so it needs more iterations.

With iterating experiments for many times, we can get the parameters range or setting strategy. The minimum number threshold value MinPts is in the range of 15–25, and the nearest neighbor threshold value $\varepsilon$ is in the range of 0.6–0.9. By calculating the cosine similarity between the mixed feature sequences representing the trajectory segments and arranging them in numerical ascending order, the nearest neighbor threshold $\varepsilon$ can be selected. The k-distance scatter diagram shown in Figure 16 is drawn, so as to select the appropriate nearest neighbor threshold $\varepsilon$. In the graph, the abscissa corresponds to the sequence number of each pair of mixed

**Figure 13.** Effects of MinPts and $\varepsilon$ on clustering results: (a) clustering with excessive MinPts (MinPts = 30), (b) low MinPts (MinPts = 2), (c) excessive $\varepsilon$ ($\varepsilon = 0.99$), and (d) low $\varepsilon$ ($\varepsilon = 0.3$).

**Figure 14.** Distribution of ano: (a) North Atlantic hurricanes and (b) synthetic trajectories.
feature sequences cosine similarity, and the ordinate is the value of cosine similarity. In order to facilitate observation and value taking, two different ranges are selected based on the ordinate to draw Figure 16(a) and (b). It can be seen from Figure 16(a) that the vertical coordinate of k-distance scatter diagram changes rapidly between 0.2 and 0.6, which makes the change trend front part of the scatter diagram unobservable. The k-distance scatter diagram shown in Figure 16(b) is drawn by removing the head points with cosine distance less than 0.6 and a small number of tail points with cosine distance greater than 0.9. It can be clearly seen that the cosine distance increases smoothly in the range of 0.6–0.9, and the cosine similarity between most of the mixed feature sequences is in this region. Therefore, the nearest neighbor threshold $\varepsilon$ is more appropriate in the range of 0.6–0.9.

In order to verify the effectiveness of our algorithm and facilitate the observation of clustering effect, it is necessary to visualize the data after clustering. Similar to Figure 13, we also use t-SNE algorithm\(^{37}\) to compress multi-dimensional data into two-dimensional data to visualize cluster results. t-SNE is especially suitable for the visualization of high-dimensional data. Figure 17 shows the effect of clustering. Figure 17(a) shows the clustering effect of $FD$ and Figure 17(b) shows the clustering effect of $FM$. It can be seen that the clustering of $FM$ is very clear to divide the trajectory segments into five clusters, which is consistent with the label in the data set, and the effect is better than that of $FD$. At the same time, it can be seen from the figure that clustering with $FM$ makes objects with similar characteristics more closely connected. Therefore, $FM$ can be a good representation of trajectory segments and comprehensively describe the shallow and deep features of trajectory segments.

There are many parameters in TAD-FM. The setting of different parameters may affect precision and recall of anomaly detection. In this article, many groups of parameters are selected to carry out multiple experiments, and the experimental results are averaged to analyze the causes of different results caused by parameter setting. The evaluation criterion used in the experiments is precision and recall. Precision refers to the proportion of positive cases in the samples identified as positive cases. Recall represents the proportion of positive cases in the sample to be correctly predicted. The two criterion are given as equations (13) and (14)

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{13}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{14}
\]

where $TP$ represents the number of samples actually positive and classified as positive by the classifier, $FP$ represents the number of samples actually negative but classified as positive by the classifier, $FN$ represents the number of samples actually positive but classified as negative by the classifier.

Table 2 shows the precision rate and recall rate of anomaly detection obtained by setting different parameters for experiments. It can be seen that different parameters will have a greater impact on the results.

\[\text{Table 2}]

| Parameter Setting | Precision | Recall |
|-------------------|-----------|--------|
| $\varepsilon = 0.6$ | 0.85      | 0.90   |
| $\varepsilon = 0.7$ | 0.90      | 0.92   |
| $\varepsilon = 0.8$ | 0.95      | 0.95   |

\[\text{Figure 15.} \text{ Loss function on synthetic trajectory data.}\]

\[\text{Figure 16.} \text{ The selection of } \varepsilon: (a) k-nn (0 < \text{cosine distance} < 1) \text{ and (b) k-nn (0.6 < \text{cosine distance} < 0.9).}\]
Too excessive $v$ will make the number of trajectory segments redundant, which is not conducive to clustering, resulting in the decline of precision and recall rate. In the case of setting the same learning rate and different iterations, the difference between the precision rate and the recall rate is not significant, because after several iterations, the difference between the two loss is very small. Too low $\varepsilon$ and too excessive MinPts directly lead to the increase in noise during clustering, which makes the result of anomaly detection inaccurate. When $\tau$ is too low, it will affect the judgment of whether the overall trajectory is anomaly, so the precision and recall rate are low.

In this article, K nearest neighbors (KNN), convolutional neural networks (CNN), support vector machines (SVM), trajectory anomaly detection with shallow feature sequence (TAD-F), trajectory anomaly detection with deep feature sequence (TAD-FD), and trajectory anomaly detection with mixed feature sequence (TAD-FM) are tested in several subsets of the data set. After many experiments, the precision of KNN is 83.96%, CNN is 87.50%, SVM is 91.29%, TAD-F is 85.46%, TAD-FD is 89.31%, and TAD-FM is 92.67%. TAD-F represents the algorithm using shallow feature sequence $F$, TAD-FD represents the algorithm using deep feature sequence $FD$, TAD-FM represents the algorithm using mixed feature sequence $FM$, that is, the unsupervised trajectory anomaly detection algorithm based on deep representation proposed in this article. KNN anomaly detection algorithm is a relatively simple detection method. By arranging the k-nearest neighbor distance of each object in descending order, the first $n$ objects can be identified as outliers. The anomaly detection method using CNN is to divide the data into training set and test set and measure the effect of anomaly detection by test set. SVM is one of the most robust and accurate two classification algorithms in data mining algorithm. The idea of using SVM for anomaly detection is to divide the objects to be detected into two categories through hyperplane, namely, normal and abnormal. Figure 18 shows the precision rate of anomaly detection. It can be clearly seen from the figure that the precision rate of TAD-FM is higher than other methods in the experimental effect. Compared with TAD-F and TAD-FD, the precision of TAD-FM is 7.21% and 2.36% higher, respectively, which shows the superiority of the deep feature fusion model based on auto-encoder and can accurately, deeply, and comprehensively serialize the trajectory sequence. As a supervised learning, CNN has a good effect when the data volume is large, but it has some limitations when the training sample data are small. In experiments using artificial trajectory data sets, the precision is low because of the small amount of data. At the same time, in the daily trajectory data, there are few labeled data, which makes it
difficult to detect the anomaly trajectory with the method of supervised learning. TAD-FM uses unsupervised method and mixed feature sequence to represent trajectories more comprehensively, which makes the measurement of similarity between trajectories more accurate, so the precision rate is higher. At the same time, the validity and feasibility of the algorithm can be verified by the labeled data set.

North Atlantic hurricanes. The North Atlantic hurricanes is a data set without label. Due to the lack of ground truth, the evaluation indicators in the field of classical machine learning cannot be applied to the evaluation of anomaly detection results. Therefore, the purpose of using the North Atlantic hurricanes data set is to verify that TAD-FM can detect anomaly trajectories in the real unlabeled situation.

Figure 19 shows the loss function of experiment, the orange (light) curve represents the mixed feature sequence $FM$, and the blue (dark) curve represents the deep feature sequence $FD$. The learning rate sets 0.5. The loss of $FM$ is gradually stable at 0.06608 and that of $FD$ is gradually stable at 0.06244. It can be roughly seen from the figure that the loss of $FM$ and $FD$ does not decrease rapidly when iterating for 100 times and 700 times, respectively, and gradually decreases slowly.

Figure 20 shows the clustering effect ($\varepsilon$: 0.7, $MinPts$: 25). In order to make the clustering of multi-dimensional data easy to observe, t-SNE algorithm is also used for dimension reduction. It can be seen from the clustering effect that the trajectory segments of DBSCAN algorithm are divided into six categories. Different from the simple method of dividing trajectory segments based on density alone, this article uses the mixed feature sequence to represent the trajectory segments of hurricanes. According to different feature sequence patterns, the trajectory segments are divided into different types, and anomalies are found in the same motion mode, which make the results more practical.

In order to verify the effectiveness of DBSCAN in this algorithm, a variety of clustering algorithms are used to compare with DBSCAN. Figure 21 shows the silhouette coefficient of K-means, Gaussian mixture model (GMM), and birch clustering algorithms with the number of clusters between 3 and 9. The performance of clustering is evaluated by silhouette coefficient. It can be seen from the figure that when the number of clusters is 6, the contour coefficient of GMM is 0.635, which is higher than other algorithms. The silhouette coefficient of DBSCAN is 0.633, which is similar to GMM. At the same time, DBSCAN does not need to specify the number of clusters, while the other three algorithms need to determine the number of clusters in advance. Considering the practical data, DBSCAN is more practical.

Table 3 shows the influence of parameters on the number of anomaly trajectories. It can be seen from the
is necessary to select appropriate parameters in the experiment. Too low \( \omega \) will make the division of trajectory segments more detailed, but too much detail will make the number of trajectory segments too redundant, which is not conducive to the analysis of anomaly trajectories. \( \varepsilon \) and \( \text{MinPts} \) directly affect the clustering effect. If \( \text{MinPts} \) is too excessive, there will be too few core sequences and too many sequences in the same cluster. At the same time, a large number of sequences that do not belong to any cluster will become noise. On the contrary, if \( \text{MinPts} \) is too low, it will produce too many core sequences, which will increase the number of clusters. Similarly, the setting of \( \varepsilon \) will also affect the clustering results. \( \tau \) considers the integrity of trajectory anomaly detection, and too excessive or too low will directly affect the number of anomaly trajectory. After many experiments, the range of nearest neighbor threshold \( \varepsilon \) is 0.6–0.9, and the minimum number threshold \( \text{MinPts} \) is 15–25.

In this article, TRAOD algorithm, TAD-FM algorithm, TAD-FD algorithm, and TAD-F algorithm are used for comparison. Due to the difference of anomaly detection methods, parameter settings, and calculation volume, they are not comparable in performance. Therefore, this article compares the experimental results. Figure 22(a) shows the experimental effect of TAD-FM algorithm. Figure 22(b) shows the experimental effect of TRAOD algorithm (\( D: 85.0, P: 0.95, F: 0.2 \)). Figure 22(c) shows the experimental effect of TAD-F algorithm. Figure 22(d) shows the experimental

### Table 3. The influence of parameters on the number of anomaly trajectories.

| \( \Omega \) | \( \varepsilon \) | \( \text{MinPts} \) | \( \tau \) | Num\(_{ano}\) |
|-----|-----|------|-----|---------|
|  125 |  0.6 |    20 |  0.5 |     63   |
|  135 |  0.7 |    25 |  0.5 |     52   |
|  135 |  0.6 |    20 |  0.7 |     41   |
|  135 |  0.7 |    25 |  0.7 |     39   |

![Figure 21. Comparison of silhouette coefficient.](image)

![Figure 22. View of trajectory anomaly detection results: (a) TAD-FM, (b) TRAOD, (c) TAD-F, and (d) TAD-FD.](image)
effect of TAD-FD algorithm. The red line is the detected anomaly trajectory. It can be seen from Figure 22 that both algorithms can detect anomaly trajectory that are significantly different from other normal trajectory, but the detection results are different. This is because the trajectory algorithm focuses on judging the anomaly from the shape of the trajectory, while the TAD-FM algorithm transforms the spatial shape of the trajectory into the feature sequence represented by the deep fusion and judges whether the trajectory is anomaly by the cosine similarity of the feature sequence. The result of TAD-F algorithm is very messy, because only using subjective characteristics, it cannot be well integrated into the algorithm. There is a certain gap between the results of TAD-FD algorithm and TAD-FM algorithm, but it is better than TAD-F algorithm. This is because there is no comprehensive fusion of all kinds of features, which cannot detect the anomaly comprehensively. Similar trajectories in shape may have different motion modes. Similarly, trajectories with different shapes may have similar characteristics. Therefore, TAD-FM algorithm can not only find the shape of the trajectory anomalies but also find the anomalies according to the hidden features, mining the anomaly trajectory in many ways, which shows the effectiveness and feasibility of the algorithm. At the same time, in the real data set, TAD-FM algorithm can also detect the anomaly trajectory, which has good practical application value.

Some real application scenarios

Trajectory anomaly detection can be widely used in numerous spatial-temporal application scenarios such as climate monitor, animal habit analysis, and traffic detection. In this section, we introduce some typical application scenarios to intuitively describe the meaning of studying trajectory anomaly detection and point out future application directions.

1. Abnormal climate detection is one of the typical application scenarios. There are mainly two ways to find abnormal climate. One is finding outlier trajectories of various storms, typhoon as well as other meteorological data. The other is finding the trends or periods of extreme climate using spatial and temporal weighted regression. Lee et al. developed a trajectory outlier detection algorithm TRAOD. Their algorithm consists of two phases: partitioning and detection. The first phase is to partition a trajectory into a set of line segments, which ensures both high quality and high efficiency. In the second phase, they present a hybrid of the distance-based and density-based approaches.

While Fotheringham et al. present a novel model called geographical and temporal weighted regression (GTWR) to account for local effects in both space and time dimensions. In their work, they first detailed both the model formulation and the estimation of GTWR focusing on spatiotemporal kernel function definition and spatiotemporal band width optimization. Then, the performance of GTWR was examined by comparison with basic geographical weighted regression models.

2. Animal habit analysis is a very important application of trajectory analysis. Li et al. developed an efficient prototype system called MoveMine to discover animal movement patterns from animal movement data. Animal abnormal habits or moving tendencies may be events or observations that do not conform to an expected pattern. These finds can greatly attract the attention from biologists or even botanists.

3. Urban traffic detection is very important for city life. Both spatial and temporal outliers can be detected from traffic data. Through analyzing trajectories in urban area or with constraint of road network, different traffic flow can be identified. Urban traffic detection is very helpful for smart city plan and smart transportation management. Li et al. proposed a temporal outlier discovery framework for detecting temporal outliers from vehicle traffic data. In their work, agglomerated temporal information of the entire data set is utilized to detect outliers. At each time stamp, similarity of each road segment is checked with other road segments, and historical similarities of each road segment are recorded in a temporal neighborhood vector. Finally, outliers are detected from drastic changes.

4. Urban safety management has been considered as a new important application of trajectory moving mining. The movement of crowds in a city is strategically important for risk assessment and public safety management. Therefore, through analyzing the status of traffic flow and communication base stations, the direction and density of moving crowds can be extracted, and quick response to the urban safety management can be provided to the city administrator. In the work of Hoang et al., a practical solution for citywide traffic prediction was developed. First, they partition the map of a city into regions using both road network and human historical mobility records. In order to model multiple complex factors that affect crowd flows, they decompose flows into three components:
seasonal patterns, trend, and residual flows. Their method is scalable and outperforms compared with some baselines.

**Conclusion**

Some current trajectory anomaly detection algorithms cannot fully take the characteristics of trajectories into account, and they are also limited by the space attribute of trajectories. In order to solve these problems, we propose an unsupervised trajectory anomaly detection algorithm based on deep representation (TAD-FM) in this article. Attributes of trajectories are deeply studied and their shallow features are extracted. Meanwhile, a deep feature fusion model based on the auto-encoder is present to obtain the mixed features of trajectories expressed by fusing both shallow and deep features. This model can comprehensively consider features of trajectories themselves. The feature sequence is used to represent trajectories and their anomalies, through which, the limitation of spatial attribute can be reduced. The experimental results show that TAD-FM algorithm has higher accuracy and can find the inconspicuous anomaly trajectories and their segments comprehensively.

As discussed above, trajectory anomaly detection is quite important for many fields. Most trajectory data are label free, and traditional classification algorithms cannot work on such data directly. Therefore, how to find anomaly using state-of-art machine learning algorithms is a challenge. Moreover, without ground-truth data, how to assess the results is another challenge.

**Acknowledgements**

The author(s) would like to thank the Research Center of Innovation on Intelligent Prevention of Disaster and Emergency Rescue (CUMT) for providing experiment facility.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the National Natural Science Foundation of China (Grant No. 71774159) and Fundamental Research Funds for the Central Universities of China (Grant No. 2015QNA38).

**ORCID iD**

Guan Yuan https://orcid.org/0000-0003-3148-9817

**References**

1. Hawkins D. *Identification of outliers*. Amsterdam: Springer, 1980.
2. Meng FR, Yuan G, Lv SQ, et al. An overview on trajectory outlier detection. *Artif Intell Rev* 2019; 52: 2437–2456.
3. Mao JL, Wang T, Jin CQ, et al. Feature grouping-based outlier detection upon streaming trajectories. *IEEE T Knowl Data Eng* 2017; 29(12): 2696–2709.
4. Zhu J, Jiang W, Liu A, et al. Effective and efficient trajectory outlier detection based on time-dependent popular route. *World Wide Web* 2017; 20: 111–134.
5. Lee JG, Han JW and Li XL. Trajectory outlier detection: a partition-and-detect framework. In: *Proceedings of the 24th IEEE international conference on data engineering*, Cancun, 7–12 April 2008, pp.140–149. New York: IEEE.
6. Yuan G, Xia SX and Zhang L. Trajectory outlier detection algorithm based on structural features. *J Comput Inform Syst* 2011; 7(11): 4137–4144.
7. Knorr EM, Ng RT and Tucakov V. Distance-based outliers: algorithms and applications. *VLDB J* 2000; 8: 237–253.
8. Breunig MM, Kriegel HP, Ng RT, et al. LOF: identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD international conference on management of data*, Dallas, TX, 16–18 May 2000, pp.93–104. New York: ACM.
9. Liu P, Pi DC and Jiang JF. Density-based trajectory outlier detection algorithm. *J Syst Eng Elec* 2013; 24(2): 335–340.
10. Chen C, Liao CW, Xie XF, et al. Trip2Vec: a deep embedding approach for clustering and profiling taxi trip purposes. *Pers Ubiquit Comput* 2019; 23: 53–66.
11. Jin W, Anthony KHT and Han JW. Mining top-n local outliers in large databases. In: *Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, CA, 26–29 August 2001, pp.293–298. New York: ACM.
12. Sutskever I, Oriol V and Quoc VL. Sequence to sequence learning with neural networks. In: *Proceedings of the 27th international conference on neural information processing systems, advances in neural information processing systems*, Montreal, QC, Canada, 8–13 December 2014, pp.3104–3112. New York: ACM.
13. Liu P, Zhang LM and Gulla JA. Real-time social recommendation based on graph embedding and temporal context. *Int J Hum-comput Syst* 2019; 121: 58–72.
14. Dong WS, Yuan T, Yang K, et al. Autoencoder regularized network for driving style representation learning. In: *Proceedings of the 26th international joint conference on artificial intelligence*, Melbourne, VIC, Australia, 19–25 August 2017, pp.1603–1609. New York: IEEE.
15. Berilli F, Capizzi G, Sciuto GL, et al. A novel training method to preserve generalization of RBPNN classifiers applied to ECG signals diagnosis. *Neural Netw* 2018; 108: 331–338.
16. Damiani ML, Hachem F, Issa H, et al. Cluster-based trajectory segmentation with local noise. *Data Min Knowl Disc* 2018; 32(4): 1017–1055.
17. Keogh E, Chu S, Hart D, et al. An online algorithm for segmenting time series. In: Proceedings of the 2001 IEEE international conference on data mining, San Jose, CA, 29 November–2 December 2001, pp.289–296. New York: IEEE.

18. Boris A, Anne D, Marc VK, et al. Segmentation of trajectories on nonmonotone criteria. ACM Algorithm 2015; 12(2): 26.

19. Maike B, Anne D, Marc VK, et al. Segmenting trajectories: a framework and algorithms using spatiotemporal criteria. J Spat Inform Sci 2011; 3: 33–63.

20. Bian J, Tian DY, Tang YY, et al. Trajectory data classification: a review. ACM T Intel Syst Tec 2019; 10(4): 1–34.

21. Sun PH, Xia SX, Yuan G, et al. An overview of moving object trajectory compression algorithms. Math Probl Eng 2016; 2016: 658730913.

22. Chang C and Zhou BY. Multi-granularity visualization of trajectory clusters using sub-trajectory clustering. In: Proceedings of the 7th IEEE international conference on data mining workshops, Miami, FL, 6 December 2009, vol. 1, pp.577–582. New York: IEEE.

23. Ossama AH, Mohamed AR, Jiang H, et al. Convolutional neural networks for speech recognition. IEEE-ACM T Audio Spe Lang Process 2014; 22(10): 1533–1545.

24. Amodei D, Anantharayanan S, Anubhai R, et al. Deep speech 2: end-to-end speech recognition in English and Mandarin. In: Proceedings of the 33rd international conference on machine learning, New York, 19–24 June 2016, pp.173–182. New York: IEEE.

25. Ulyanov D, Vedaldi A and Lempitsky V. Deep image prior. In: Proceedings of the 2018 IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, 18–23 June 2018, pp.9446–9454. New York: IEEE.

26. Wang MX, Lu ZD, Zhou J, et al. Deep neural machine translation with linear associative unit, 2017, https://arxiv.org/abs/1705.00681

27. Dong WS, Li J, Yao RJ, et al. Characterizing driving styles with deep learning, 2016, https://arxiv.org/abs/1607.03611

28. Kyunghyun C, Van Merriboer B, Bahdanau D, et al. On the properties of neural machine translation: encoder-decoder approaches, 2014, https://arxiv.org/abs/1409.1259

29. Wu JL, Long KY, Wang F, et al. Deep comprehensive correlation mining for image clustering. In: Proceedings of the 2019 IEEE international conference on computer vision, Seoul, 27 October–2 November 2019, pp.8150–8159. New York: IEEE.

30. Ester M, Kriegel HP, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd international conference on knowledge discovery and data mining, Portland, OR, 2–4 August 1996, pp.226–231. New York: ACM.

31. Han J, Kamber M and Pei J. Data mining: concepts and techniques, 3rd ed. New York: Elsevier, 2012, pp.543–584.

32. Dai B, Helmut K and Andrea K. Segmenting trajectories by movement states. In: Sabine T and Patrick L (eds) Advances in Spatial Data Handling. Berlin: Springer, 2013, pp.15–25.

33. Damiani ML, Issa H, Fotino G, et al. Introducing “presence” and “stationarity index” to study partial migration patterns: an application of a spatio-temporal clustering technique. Int J Geogr Inf Sci 2016; 30(5): 907–928.

34. http://weather.unisys.com/hurricanes (available 5 November 2019).

35. http://avires.dimi.uniud.it/papers/trclust/ (available 1 November 2019).

36. Piciarelli C, Micheloni C and Foresti GL. Trajectory-based anomalous event detection. IEEE T Circ Syst Vid 2008; 18(11): 1544–1554.

37. Jiang SY, Li QH, Li KL, et al. GLOF: a new approach for mining local outlier. In: Proceedings of the 2003 IEEE international conference on machine learning and cybernetics, Xi'an, China, 2–5 November 2003, pp.157–162. New York: IEEE.

38. Fotheringham AS, Crespo R and Yao J. Geographical and temporal weighted regression. Geogr Anal 2015; 47: 431–452.

39. Li ZH, Han JW, Ji M, et al. MoveMine: mining moving object data for discovery of animal movement patterns. J ACM Trans Intell Syst Technol 2011; 2(4): 1–37.

40. Li XL, Li ZH, Han JW, et al. Temporal outlier detection in vehicle traffic data. In: Proceedings of the 25th IEEE international conference on data engineering, Shanghai, China, March 2009, pp.1319–1332. New York: IEEE.

41. Hoang MX, Zheng Y and Singh AK. FCCF: forecasting city wide crowd flows based on big data. In: Proceedings of the 24th international conference on advances in geographic information systems, Burlingame, 31 October–3 November 2016, pp.1–10. New York: IEEE.