IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 May 2023, accepted 14 June 2023, date of publication 12 July 2023, date of current version 12 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294689

RESEARCH ARTICLE

Transfer Learning for Region-Wide Trajectory Outlier Detection

YUEYANG SU[1,2], (Student Member, IEEE), DI YAO[1,2], (Member, IEEE), TIAN TIAN[3], (Member, IEEE), AND JINGPING BI[1,2], (Member, IEEE)

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 101408, China
[3]Nanjing Marine Radar Institute, Nanjing 210000, China

Corresponding author: Jingping Bi (bjp@ict.ac.cn)

This work was supported in part by the NSFC under Grant 62002343 and Grant 6207704.

ABSTRACT Trajectory outlier detection is a crucial task in trajectory data mining and has received significant attention. However, the distribution of trajectories is tied to social activities, resulting in extreme unevenness among regions. While existing methods have demonstrated excellent performance in regions with sufficient historical trajectories, they frequently struggle to detect outliers in regions with limited trajectories. Unfortunately, this issue has not received much attention, leaving a gap in the current understanding of trajectory mining. To deal with this problem, we in this paper propose a model called TTOD that can effectively detect outliers in regions with sparse data by transferring knowledge among regions. The main idea is to learn a feature mapping function that maps the global feature space of auxiliary regions to the target region's specific feature space. To achieve this, we adopt a VAE-based model called the Global VAE to learn the global feature space in auxiliary regions by modeling the trajectory patterns with Gaussian distributions. Then, we propose a Specific-region VAE that serves as the mapping function to learn the target feature space. Additionally, considering the data drift of feature distributions among regions, we introduced an additional pattern synthesis layer, named the De-drift Layer, to diversify the target feature space, thus addressing the pattern missing issue caused by the gap of feature distributions between the auxiliary regions and the target regions. Then the target feature space can be well studied and applied to detect outliers. Finally, we conduct extensive experiments on two real taxi trajectory datasets and the results show that TTOD achieves state-of-the-art performance compared with the baselines.

INDEX TERMS Trajectory outlier detection, transfer learning, VAE, spatial-temporal data, trajectory data mining.

I. INTRODUCTION

Benefiting from the rapid development of sensing technologies such as Global Positioning System (GPS) and road sensors, collecting trajectories is no longer a labor-intensive task, which makes it possible to analyze the moving targets' behaviors based on trajectories [1], [26], [52], [53], [54], [55], [56]. For example, in urban areas, a significant number of taxi trajectories are generated daily, and they can be effortlessly collected through road sensors and drivers' mobile phones. This vast pool of data provides ample resources to analyze the driving behaviors of taxis.

As one of the essential research branches in trajectory data mining, trajectory outlier detection has received significant attention and related techniques have been widely applied to various fields, such as surveillance, traffic planning, wildlife protection, and *etc.* [2]. Specifically, Trajectory outliers [3], [4], [23] refer to the trajectories or trajectory segments that deviate significantly from the expected or typical movement patterns, such as detour, abrupt speed changes, and *etc.*. Identifying and analyzing these outliers can provide valuable insights into the behavior and characteristics of the moving targets, enabling researchers to detect potential

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

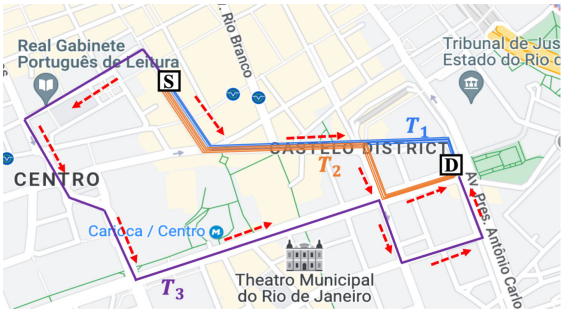


FIGURE 1. The example of taxi driving fraud. The T_1 and T_2 are normal trajectories, the T_3 is outlier.

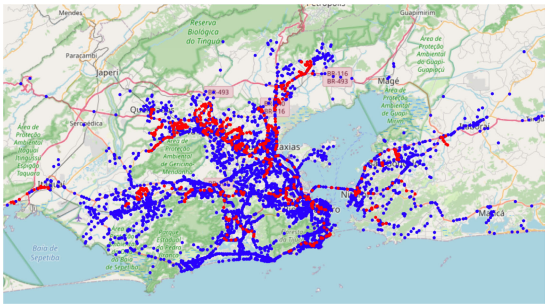


FIGURE 2. The spatial distribution of taxi trajectories in Rio de Janeiro. The blue points are the normal trajectories and the red are the outliers.

problems early and take appropriate measures to mitigate their effects. Given the issue of taxi driver fraud, as shown in Figure 1, the T_1 and T_2 follow the mainstream paths and are considered normal trajectories. However, some drivers may intentionally take a longer detour(*e.g.* T_3) from the mainstream paths to increase the fare which is harmful to the interests of passengers. Detecting such detour trajectories is of great importance for the safety of passengers' lives and property.

There has been a lot of research [2], [5], [6], [7], [57], [58] on trajectory outlier detection and existing methods can be divided into two categories, *i.e.*, metric-based methods and learning-based methods. The metric-based methods [8], [9], [13] identify anomalous trajectories based on their distance from other trajectories or reference trajectories. However, these methods rely on artificial features, which have limited expressiveness and may not be able to capture the deep patterns of trajectories. The learning-based methods [17], [21] tend to detect anomalous trajectories with machine learning models such as VAE [48], LSTM [49], Transformer [50], and *etc.*, which is able to model the complex spatio-temporal features in trajectories and mine the latent patterns of movements. As a result, they have gained widespread popularity in both academic research and industrial applications, and have become the mainstream approaches for trajectory outlier detection. However, both supervised and unsupervised learning methods are large-scale data-demanding, which may not always be met due to the skewed distribution of trajectories.

Skewed distribution of trajectories [24], [25] refers to the situation where the trajectories are unevenly distributed across the available time or space. This can be attributed to several reasons, such as the natural variability of human movement patterns, population density, and *etc.*. Generally speaking, urban regions tend to have higher population densities and more points of interest than rural regions, leading to more trajectories. Moreover, people tend to repeatedly follow the same paths when engaging in a particular activity (*e.g.* going to work or returning home), which may result in a concentration of trajectories along those paths, also creating a skewed distribution. For example, the spatial distribution of taxi trajectories in Rio de Janeiro, as shown in Figure 2, reveals that the majority of the trajectories are concentrated in the city center while there are relatively few in the suburbs. This case highlights a notable disparity in the trajectories amounts between different regions.

According to the scales of available data, the space can be further separated into data-rich and data-sparse regions. For data-rich regions, many existing methods have been proven to be effective with sufficient history trajectories. However, in data-sparse regions, these models struggle to converge effectively. To solve this problem, the intuitive idea [36], [37], [38], [39] is to leverage models trained in data-rich regions to detect trajectory outliers in data-sparse regions. However, there is a significant analytical bias caused by the data drift among regions, which may lead to poor performance. Specifically, the spatio-temporal characteristics differ greatly among regions, which means that there is a great data drift of the feature distributions between the trajectories in data-rich and data-sparse regions. As a result, it is challenging to generalize a model trained with the trajectories in data-rich regions to detect the outliers in data-sparse regions. For example, if a model is trained on trajectories in a region with a dense urban environment and high traffic congestion, it may not perform well when directly applied to a rural area with low traffic density and different road structures. The model may fail to capture the unique features of the target region, such as different traffic patterns and road types.

To tackle these challenges, we proposed a Transfer-learning method for Trajectory Outlier Detection, namely TTOD, to detect outliers in data-sparse regions (*i.e.* the region-wide trajectory outlier detection with sparse data). For the data sparsity problem, we aim to extract the global knowledge from auxiliary regions and model the feature distribution with Gaussian distribution to enhance the detection task in target regions. Instead of transferring from one region to another, the global feature space learned from the several auxiliary regions is expected to contain diverse spatio-temporal features and potential path patterns. Then the feature space of target regions can be considered as a subspace of it with some drift. As shown in Figure 3, the red circle corresponds to the global feature space of the trajectory data in auxiliary regions, while the blue circle represents the feature space of the target region. Furthermore, the target region's feature space can be partitioned into two distinct parts. The first is the specific-subspace, which is a subspace of the global feature space.

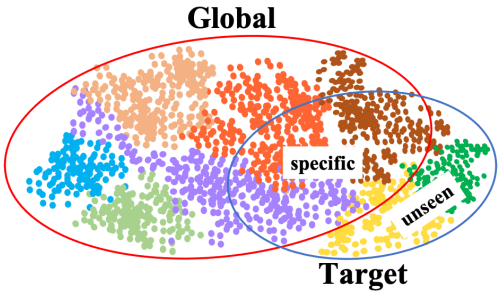


FIGURE 3. Illustration of the feature space, where different colored dots represent distinct patterns. The red circle represents the global feature space, while the blue circle represents the target feature space consisting of specific-subspace and unseen-subspace.

The second is the unseen-subspace, an unknown feature space that arises due to the data drift of feature distributions. Our TTOD aims to learn the mapping function from the global feature space to the specific-subspace for knowledge transferring. To deal with the data drift problem, we introduced a De-drift Layer that synthesizes previously unseen feature patterns, thereby approximating the unseen-subspace. Moreover, the proposed layer can also prevent the model from overfitting to the feature distributions in auxiliary regions, enabling better generalization to the target regions.

Specifically, as shown in Figure 5, our TTOD is a VAE-based model with a nested structure, which consists of the Global VAE, the Specific-region VAE, and the De-drift Layer. The Global VAE is designed to model the global feature space with Gaussian mixture distribution using trajectories in the auxiliary regions. Then the Specific-region VAE is leveraged to work in the global feature space and act as a mapping function between the global feature space and the specific-subspace of target regions. To synthesizes the unseen feature patterns, the De-drift Layer takes the learned target feature distributions as input and transforms with a parameterized Gaussian distribution to approximate the unseen-subspace. Then, the latent space of the Specific-region VAE is able to converge to the target feature space. Finally, by combining the Global VAE encoder, the Specific-region VAE encoder, and the De-drift Layer, we can effectively model trajectories in the target regions. Additionally, we added an extra classifier to detect outliers. It is noteworthy that a feature space with higher expressiveness is more likely to be effective in terms of generalizing to the target regions. To enhance the expressiveness of the learned global feature space, we make use of deep networks as encoders and decoders. Conversely, the Specific-region VAE is implemented using a lightweight network so that it can be trained effectively with limited trajectories in the target regions.

Overall, the main contributions are summarized as follows:

- To the best of our knowledge, this is the first work that focuses on the data drift problem of feature distributions among regions in region-wide trajectory outlier detection with sparse data.
- We proposed a novel model, TTOD, for region-wide trajectory outlier detection. Our TTOD has the unique ability to transfer global knowledge among regions and synthesize unseen feature patterns, resulting in a strong performance in the target regions with limited trajectories.
- Extensive experiments on two real taxi trajectory datasets show that TTOD achieves state-of-the-art performance.

II. RELATED WORK

Trajectory outlier detection has received significant attention and a wide range of methods [6], [7], [15], [20], [27], [28] have been proposed. We divide the existing methods into two categories, including metric-based methods and learning-based methods. The details about methods can be found in the following subsections.

A. METRIC-BASED METHODS

Metric-based methods are utilized to detect outliers using distance metrics that are designed for specific tasks. These methods identify trajectories that exhibit significant dissimilarity compared to the rest or the reference trajectories as outliers. Lee et al. [6] proposed a framework to detect sub-trajectory outliers. The algorithm first divides the trajectories into t-partitions to capture the fine-grained details. And then assigns a score to each t-partition by calculating the sum of densities of t-partitions from all observed trajectories within the same time window. If the score of a particular t-partition exceeds 1, it is identified as an outlier, indicating a significant deviation from the normal patterns observed in the data. Liu et al. [8] introduced a novel framework called RTOD. The approach involves dividing trajectories into segments and computing the distance between segments using the Hausdorff distance metric. Outlier trajectories are then identified based on their relative distance scores. Saleem et al. [9] presented an algorithm named RPAT. They partitioned trajectories into sub-trajectories based on road segments and computed scores for each sub-trajectory using handcrafted features. Then the trajectories exceeding a predefined threshold are flagged as outliers. Zhu et al. [10] proposed a time-dependent transfer graph for each group of trajectories with the same source and destination, and identified the top-k most popular routes as reference routes for each time period. An incoming trajectory is flagged as an outlier if it differs significantly from the reference routes in both spatial and temporal dimensions.

Some researchers also leverage clustering algorithms [11], [12], [29], [30] to group similar trajectories and then identify the trajectories that belong to the clusters with few members as outliers. Ying et al. [12] introduced a novel similarity measure that incorporates both spatial and temporal dimensions of trajectories using minimal bounding boxes (MBBs). They proposed to cluster the trajectories with the DBSCAN algorithm and identify the trajectories in clusters with a lower density as outliers. Wang et al. [11] focused on detecting anomalous taxi trajectories and developing a method based on edit distance and hierarchical clustering.

They took the taxi trajectories with the same source and destination pairs as a group and used an edit distance algorithm to measure the similarity between them. Then an adaptive hierarchical clustering algorithm is proposed to differentiate anomalous trajectories from normal ones. Zhang et al. [14] proposed to use the feature-based DBSCAN algorithm to label the trajectory partitions. And then trained the classical classification models such as SVMs with the labeled trajectories to detect outliers in a stream way. Huang et al. [15] proposed a KNN-based method to detect ship anomaly behavior. The approach involves filtering the ship anomaly data candidate set using the KNN algorithm and calculating the local deviation index using the LOF algorithm. Then, they detected the ship anomaly behaviors with a predefined threshold.

However, these methods are limited in the weak expression of artificial features and can not capture the deep patterns in trajectories. Additionally, they are sensitive to the proposed metrics which vary across application domains, leading to potential inconsistency in performance.

B. LEARNING-BASED METHODS

Some researchers proposed leveraging machine learning models in trajectory outlier detection. These methods can be broadly classified into two categories: supervised learning and unsupervised learning methods.

Supervised learning methods [17], [31], [32] have gained significant popularity in trajectory outlier detection. These approaches involve training models using datasets with labeled trajectories, which enables them to learn patterns and characteristics associated with normal or outlier behaviors. Once trained, the models can be deployed to effectively identify outliers. Song et al. [16] presented a model called ATD-RNN, which employs a stacked RNN neural network to capture sequential information and internal characteristics that differentiate anomalous from normal trajectories. Then they detected outliers using a Multi-Layer perceptron. Sillito et al. [17] proposed a novel model for analyzing pedestrian behavior by representing their trajectories using approximating cubic spline curves. The method utilizes an incremental semi-supervised learning procedure and provides a more interactive learning experience between the system and human operators. Cheng et al. [18] introduced a model called ST-RNN. This approach employs the RNN to capture the underlying structure features of the trajectories and utilized an attention mechanism to focus on the most relevant spatial-temporal information.

Given the labor-intensive process of labeling trajectories, a growing number of researchers have turned their attention towards exploring unsupervised learning methods [33], [34], [35]. These methods aim to learn patterns and detect outlier trajectories without the demands for pre-labeled data. Gray et al. [19] proposed a GAN-based model to detect trajectory outliers. They employed an infinite Gaussian mixture model in combination with bi-directional generative adversarial networks to detect outliers in the latent space by using a multi-modal Mahalanobis metric. Wu et al. [20] introduced a model called DB-TOD to detect outliers of vehicle trajectories. They utilized a probabilistic model to capture driving behavior and preferences in unlabeled historical trajectories. Then the model employs explicit feature counts and latent feature biases to calculate the latent cost of routing decisions, which facilitates the detection of outliers. Liu et al. [21] proposed a VAE-based model for online trajectory outlier detection, called GM-VSAE. They proposed to model the probability distribution of the route patterns in the latent space with a Gaussian mixture distribution, which enables the discovery of different types of normal routes and facilitates effective trajectory outlier detection. Han et al. [22] presented a time-dependent anomaly detection model called DeepTEA, which aims to identify anomalous movements of vehicles on the roads network. They introduced a CNN to learn the traffic conditions and utilized a Gaussian Mixture Variational Autoencoder to learn latent patterns of trajectories. Then the learned patterns are leveraged to identify time-dependent outliers.

Whether it is a supervised learning method or an unsupervised learning method, both require a large amount of labeled or unlabeled data, making them unsuitable for trajectory anomaly detection with limited data.

TABLE 1. Notations used in this paper.

Notations	Descriptions
$T, (S-D)$	A trajectory, the source and destination of a trajectory.
$p_i = (x_i, y_i, t_i)$	A GPS point indicates the latitude x_i and longitude y_i of the moving object at t_i.
e_i	A road segment number in road network.
$R_i, T_{R_i}, \mathcal{R}$	A region in city, history trajectories in region R_i, a set of regions.
$G(V, E)$	Node graph of the road network
$G'(V', E'), \mathbf{A}, \mathbf{X}$	Edge graph of the road network, the adjacent matrix of G', the node attributes of G'.
$\mathbf{E}$	The learned embedding matrix of road segments.
$\phi = \{\phi_e, \phi_\mu, \phi_\sigma, \phi_d, \phi_{out}\}$	The parameters in the Global VAE.
$\Theta = \{\phi_e^{'}, \phi_d^{'}, \phi_\mu^{'}, \phi_\sigma^{'}\}$	The parameters in the Specific-region VAE.
$\theta^\sigma = \{\theta_s^\sigma, \theta_b^\sigma)\}$	The parameters in the De-drift Layer.
$\mathcal{L}_{Glob}, \mathcal{L}_{rec}^g$	The total loss and the reconstruction loss of the Global VAE.
$\mathcal{L}_{spec}, \mathcal{L}_{rec}^s$	The total loss and the reconstruction loss of the Specific-region VAE.

III. PRELIMINARY

A. NOTATIONS AND CONCEPTS

In this section, we introduce the key concepts used throughout this paper below and provide main notations in Table 1 for reference.

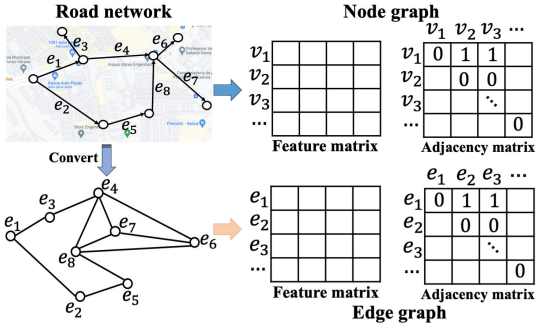


FIGURE 4. The process of road network modeling.

1) RAW TRAJECTORY

A raw trajectory is a sequence of GPS points that are ordered by time, denoted as $T =< p_1, \cdots, p_i, \cdots, p_n >$. Each point p_i is a three tuple (x_i, y_i, t_i), representing the latitude x_i and longitude y_i of the moving object at timestamp t_i. The length of the trajectory T is n, which indicates the number of points in the sequence.

2) ROAD NETWORK

A road network is typically depicted as a directed node graph $G(V, E)$, where the set of vertices V corresponds to crossroads or intersections, and the set of edges E represents the road segments connecting these vertices.

3) MAPPED TRAJECTORY

A mapped trajectory is a time-ordered sequence of road segments, represented by $T =< e_1, \cdots, e_i, \cdots, e_m >$. And each e_i corresponds to a specific road segment number on the road network.

4) TRAJECTORY OUTLIER

For a given source-destination $(S-D)$ pair, there are typically popular routes that the majority of trajectories follow. Trajectories that conform to these routes are considered normal, whereas those that deviate from these popular routes are outliers.

B. PROBLEM DEFINITION

1) REGION-WIDE TRAJECTORY OUTLIER DETECTION

Given a set of auxiliary regions $\mathcal{R} = \{R_1, R_2, \cdots, R_P\}$, along with their historical trajectories $\mathcal{T}_{R_{1:P}}$. $\mathcal{T}_{R_i}$ represents the unlabeled history trajectories in R_i. Our objective is to identify the outliers in a target region $R_{tar} \notin \mathcal{R}$ which characterized by sparse labeled trajectories $\mathcal{T}_{R_{tar}}$.

IV. METHODOLOGY

In this section, we specify the data pre-processing and the details of our proposed TTOD, respectively.

A. DATA PRE-PROCESSING

Since the trajectory outliers are related to spatio-temporal environments, we introduce road information to model environmental factors and enhance the detection performance. Specifically, for a given city, we extract the road information, including both the map structure and road properties (*e.g.* road category, road length, speed limits, and *etc.*), from OpenStreetMap[1]. The structure information can be used to build the road network, while the road properties can provide valuable insights to the traffic environment. To model such road properties, we further convert the node graph of road networks $G(V, E)$ to an edge graph $G'(V', E')$. The $V' = \{e_1, \cdots, e_i, \cdots, e_N\}$ is a set of roads consisting of N road segments and E' is a set of crossroads and intersections. To represent the connection between roads, we use an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, which is an $N \times N$ matrix. A value of 1 in A_{ij} indicates that road i and j are connected, while a value of 0 indicates that they are not connected. Furthermore, to model the traffic patterns, we incorporate the average speed derived from history trajectories, along with the road properties, as attributes of the nodes. We denote the attributes as $\mathbf{X} \in \mathbb{R}^{N \times P}$, which is an $N \times P$ matrix, where P is the number of features. The process is illustrated in Figure 4.

As common sense, the spatio-temporal states of road segments in a local area are similar. Firstly, road segments in a local area typically share similar environments, which can impact the speed and density of vehicles, and consequently, the traffic conditions of the road segments. Secondly, traffic flow is a continuous process, and the speed and density of vehicles on one road segment can influence the flow on adjacent road segments. For instance, When a road segment is heavily congested, causing delays and potentially leads to further congestion on neighboring road segments, resulting in similar spatio-temporal states in the local area. So we introduce GCN [51] to learn the roads embedding on graph G' and enhance the embedding by message-passing among neighbors. Following the methods described in [46], we trained the GCN with a self-supervised pre-training task that aims to enhance the mutual information between a node's hidden representation and those of its neighbors. By taking this approach, we encourage each node representation to capture and incorporate contextual information from its surrounding neighbors representations. Further details about the method can be found in [46]. As a result, we can learn an embedding matrix $\mathbf{E} \in \mathbb{R}^{d_e \times N}$ of road segments, denoted as:

$$\mathbf{E} = \mathrm{GCN}(G', \mathbf{X}, \mathbf{A}) \tag{1}$$

where d_e is the dimension of road segment embedding and N is the total number of road segments.

Given a mapped trajectory $T =< e_1, \cdots, e_i, \cdots, e_L >$, we can obtain the embedding of each road segment by $\mathbf{e}_i = \mathrm{Embedding}(e_i, \mathbf{E})$. Then the resulting processed sequence T_r can be denoted as $T_r =< \mathbf{e}_1 \cdots \mathbf{e}_L >$.

B. MODEL ARCHITECTURE

In this subsection, we describe the details of our proposed TTOD. As shown in Figure 5, TTOD is a VAE-based

[1] https://www.openstreetmap.org/

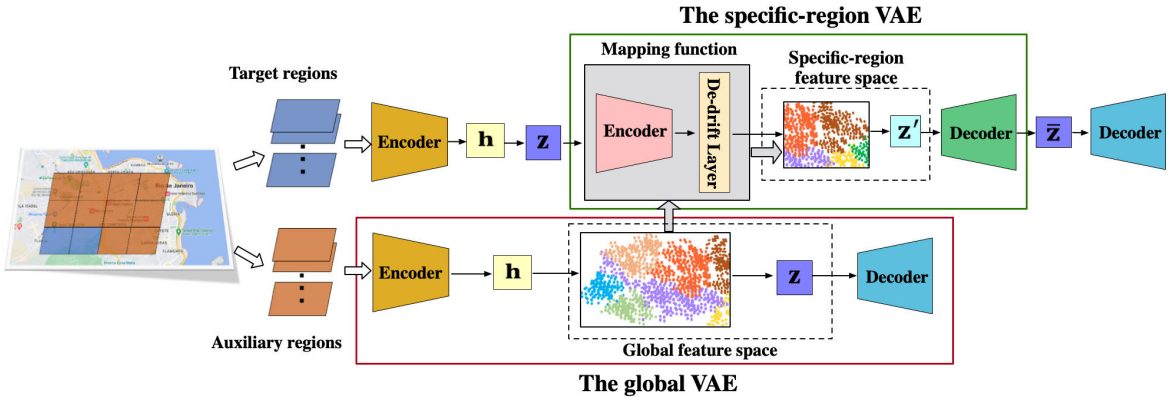


FIGURE 5. The model architecture of TTOD.

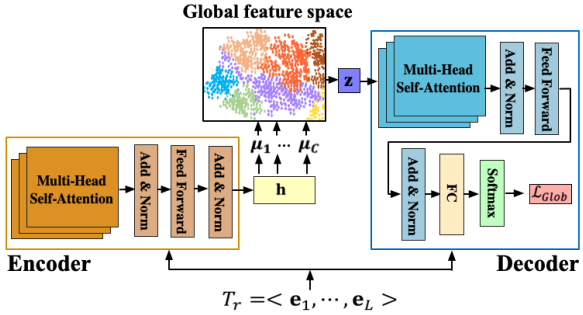


FIGURE 6. The model architecture of the Global VAE.

model and consists of three components: the Global VAE, the Specific-region VAE, and the De-drift Layer. We will describe each component in detail below.

1) THE GLOBAL VAE

To better identify the outlier trajectories in regions with sparse data, we seek to extract the global knowledge from trajectories in auxiliary regions to enhance the detection in the target region. To accomplish this, we introduce the Global VAE. As shown in Figure 6, it is designed to model the global feature space with a Transformer-based VAE and trained on trajectories in auxiliary regions. We define the global feature space as a collection of spatio-temporal features and potential path patterns, which is expected to be diverse and semantically rich. As a result, we have opted to use a Gaussian mixture distribution to model the space.

Specifically, we first leverage a Transformer-based encoder to capture the spatio-temporal features among the road segments in trajectories and learn the trajectory embedding. Formally, let L denote the length of the sequence, then the Transformer-based encoder takes $T_r =< \mathbf{e}_1 \cdots \mathbf{e}_L >$ as input and generates the latent representation $\mathbf{h}_T$ with the following formulas:

$$\mathbf{h}_T = \mathrm{Enc}_{TRFM}(\mathbf{e}_{i...L}, \phi_e) \tag{2}$$

where $\mathrm{Enc}_{TRFM}(\cdot)$ is the Transformer-based encoder and the ϕ_e are its parameters.

Following the Transformer-based encoder above, we introduce a Gaussian mixture distribution to model the diverse global feature space. Specifically, we leverage multinomial distribution to model the different types of spatio-temporal features and path patterns, the prior probability distribution can be denoted as:

$$p_\theta(c) = \mathrm{Mult}(\boldsymbol{\pi}) \tag{3}$$

where $\boldsymbol{\pi} \in \mathbb{R}^C$, C is the number of types and $\Sigma_{i=1}^{C} \boldsymbol{\pi}_i = 1$. To elaborate further, we employ the Gaussian function to describe the feature distribution of each type. The c-th type's Gaussian distribution can be mathematically represented as:

$$p_\theta(\mathbf{z}|c) = \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2 \mathrm{I}) \tag{4}$$

where $\boldsymbol{\mu}_c \in \mathbb{R}^d, \boldsymbol{\sigma}_c \in \mathbb{R}^d$ are the mean and standard deviation vectors, d is the dimension of the feature space. Finally, the latent feature space can be modeled as:

$$p_\theta(\mathbf{z}) = p_\theta(\mathbf{z}|c)p_\theta(c) \tag{5}$$

Supposed that the latent feature distribution can be fitted by C Gaussian distributions, then for a given trajectory T, the latent variable drawn from the posterior distribution can be denoted as:

$$\begin{aligned} \mathbf{z}_T \sim q_\phi(\mathbf{z}, c|T) &= q_\phi(\mathbf{z}|T)q_\phi(c|T) \\ q_\phi(\mathbf{z}|T) &= \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\sigma}_T^2 \mathrm{I}) \\ \boldsymbol{\mu}_T &= \mathrm{g}_1(\mathbf{h}_T, \phi_\mu) \\ \boldsymbol{\sigma}_T &= \mathrm{g}_2(\mathbf{h}_T, \phi_\sigma) \\ q_\phi(c|T) := p_\theta(c|\mathbf{z}_T) &= \frac{p_\theta(c)p_\theta(\mathbf{z}_T|c)}{\sum_{i=1}^{C} p_\theta(c_i)p_\theta(\mathbf{z}_T|c_i)} \end{aligned} \tag{6}$$

where $\boldsymbol{\mu}_T \in \mathbb{R}^d$, $\boldsymbol{\sigma}_T \in \mathbb{R}^d$ are the mean and standard deviation vectors of the Gaussian distribution. $\mathrm{g}_1(\cdot)$ and $\mathrm{g}_2(\cdot)$ are the fully connected layers.

To optimize the global feature space, we introduce a trajectory reconstruction task, which is accomplished by a Transformer-based decoder. Specifically, considering an ongoing trip, the selection of the next road segments depends not only on the observed trajectory but also on the destination

information. According to this observation, we introduce a sequential generation schema that predicts the next road segments while taking the destination into consideration, the process can be denoted as:

$$\begin{aligned} &\bar{\mathbf{h}}_i = \mathrm{Dec}_{TRMF}(\mathbf{e}_{des}, \bar{\mathbf{h}}_{i\text{-}1}, \phi_d) \\ \textit{where} \quad &i = 1, 2, \cdots, L \quad \bar{\mathbf{h}}_0 = \mathbf{z}_T \sim q_\phi(\mathbf{z}, c|T) \\ &\bar{\mathbf{e}}_i = \mathrm{f}_1(\bar{\mathbf{h}}_i, \phi_{out}) \\ &\bar{e}_i \sim p_\theta(\bar{e}|\bar{\mathbf{e}}_{i-1}) = \mathrm{Mult}(\mathrm{softmax}(\bar{\mathbf{e}}_{i-1})) \end{aligned} \tag{7}$$

where the $\bar{e}_i$ is the reconstructed road segments number, $\mathbf{e}_{des}$ is the road embedding of destination, and the $\bar{\mathbf{h}}_{i\text{-}1}$ is the hidden vector of the previous sequence $Step_{<i}$. $f_1(\cdot)$ maps $\bar{\mathbf{h}}_i$ to a vector with N dimension, which is the same as the number of road segments.

Similar to the general VAE, we define the loss function for the Global VAE as the sum of a reconstruction loss and a Kullback-Leibler (KL) loss. The reconstruction loss is calculated with the cross-entropy function(CE) between the reconstructed $\bar{e}_i$ and the original road segment e_i, denoted as: $\mathcal{L}^g_{rec}(\bar{e}_i, e_i) = \mathrm{CE}(\bar{e}_i, e_i)$. Then the total loss can be represented as:

$$\begin{aligned} \mathcal{L}_{Glob} = \Sigma_{i=1}^{L} \mathcal{L}^g_{rec}(\bar{e}_i, e_i) + \mathrm{KL}[q_\phi(c|T) \| p_\theta(c)] \\ + \mathrm{KL}[q_\phi(\mathbf{z}|T) \| p_\theta(\mathbf{z}|c)] \end{aligned} \tag{8}$$

where KL[., .] is the Kullback-Leibler function.

2) THE SPECIFIC-REGION VAE

After obtaining the global feature space with the Global VAE, we expect to establish a link between it and the target feature space to enable knowledge transfer. To accomplish this, we introduce an embedded VAE called the Specific-region VAE, which is designed to learn a mapping function from the global feature space to the specific-subspace of the target regions' feature space.

As shown in Figure 5, the Specific-region VAE is an MLP-based model with a lightweight architecture, which works in the latent space of the Global VAE. We utilize a single Gaussian distribution to model the target feature space while considering training stability. Specifically, given a processed trajectory $T_r =< \mathbf{e}_1 \cdots \mathbf{e}_L >$ from the target region, we first obtain the corresponding latent variable $\mathbf{z}_T$ with the learned encoder Enc_{TRFM} of the Global VAE, and take it as the input of the Specific-region VAE. Then a MLP-based encoder is leveraged to process the $\mathbf{z}_T$ into the latent embedding $\mathbf{h}'_T$ in the target feature space. The operation can be denoted as:

$$\begin{aligned} \mathbf{z}_T \sim q_\phi(\mathbf{z}, c|T) = q_\phi(\mathbf{z}|T) q_\phi(c|T) \\ \mathbf{h}'_T = \mathrm{Enc}_{MLP}(\mathbf{z}_T, \phi'_e) \end{aligned} \tag{9}$$

where $\mathrm{Enc}_{MLP}(\cdot)$ is the MLP-based encoder of the Specific-region VAE. Then the posterior distribution of the target feature space can be denoted as:

$$\mathbf{z}'_T \sim q_{\phi'}(\mathbf{z}'|\mathbf{z}_T) = \mathcal{N}(\boldsymbol{\mu}'_T, {\boldsymbol{\sigma}'_T}^2 \mathrm{I})$$

$$\begin{aligned} \boldsymbol{\mu}'_T = \mathrm{g}'_1(\mathbf{h}'_T, \phi'_\mu) \\ \boldsymbol{\sigma}'_T = \mathrm{g}'_2(\mathbf{h}'_T, \phi'_\sigma) \end{aligned} \tag{10}$$

where $\mathbf{z}'_T$ is the latent variant in the target feature space. $\boldsymbol{\mu}'_T \in \mathbb{R}^{d_{tar}}$, $\boldsymbol{\sigma}'_T \in \mathbb{R}^{d_{tar}}$ are the mean and standard deviation vectors of Gaussian distribution in the target feature space with d_{tar}-dimension. $\mathrm{g}'_1(\cdot)$ and $\mathrm{g}'_2(\cdot)$ are the fully connected layers.

Then a MLP-based decoder is introduced to reconstruct the $\mathbf{z}_T$, the operation can be denoted as:

$$\bar{\mathbf{z}}_T = \mathrm{Dec}_{MLP}(\mathbf{z}'_T, \phi'_d) \tag{11}$$

where $\bar{\mathbf{z}}_T$ is the reconstruction of $\mathbf{z}_T$. Then we leverage Mean Squared Error(MSE) to calculate the reconstruction loss, *i.e.*, $\mathcal{L}^s_{rec}(\bar{\mathbf{z}}_T, \mathbf{z}_T) = \mathrm{MSE}(\bar{\mathbf{z}}_T, \mathbf{z}_T)$. Similar to the Global VAE, the total loss can be defined as:

$$\mathcal{L}_{Spec}(\mathbf{z}_T) = \mathcal{L}^s_{rec}(\bar{\mathbf{z}}_T, \mathbf{z}_T) + \mathrm{KL}(q(\mathbf{z}'_T|\mathbf{z}_T) || p_{\theta'}(\mathbf{z}'_T)) \tag{12}$$

where $p_{\theta'}(\mathbf{z}'_T)$ is the prior normal distribution of the target feature space. Notation that our Specific-region VAE is optimized with the sparse trajectories in the target region, and we have fixed the parameters of the Global VAE to prevent overfitting to the dataset during this stage.

3) DE-DRIFT LAYER

With the Specific-region VAE, we can obtain the specific-subspace of the target feature space. However, as discussed in section I, there is a gap between the feature space of the auxiliary regions and the target region, *i.e.*, the unseen-subspace, which is caused by the data drift. We aim to eliminate the gap by further expanding the specific-subspace of the target feature space with a De-bias Layer, so as to approximate the unseen-subspace. As shown in Figure 6, our De-drift layer can be viewed as a transformation that follows the encoder of the Specific-region VAE. This layer utilizes a pair of parameters, namely scaling and bias, to expand the diversity of the feature distributions in the target region.

Specifically, in our approach, we draw the scaling($\boldsymbol{s}$) and the bias($\boldsymbol{b}$) from Gaussian distributions, respectively. The operation can be denoted as:

$$\boldsymbol{s} \sim \mathcal{N}(1, \mathrm{softplus}(\theta^\sigma_s)), \boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, \mathrm{softplus}(\theta^\sigma_b)) \tag{13}$$

where θ^σ_s and θ^σ_b are the standard deviations, softplus$(\cdot)$ are the softplus function. Regarding the trajectory representation $\mathbf{z}'_T$, which is produced by the encoder of the Specific-region VAE, the transformation operation can be formulated as follows:

$$\mathbf{z}'^{\,transf}_T = \mathbf{z}'_T \times \boldsymbol{s} + \boldsymbol{b} \tag{14}$$

Following the above operation, the target feature space can be expanded. However, there is still one problem left, *i.e.*, how to obtain the parameters θ^σ_s and θ^σ_b. To solve this problem, we redesign the optimization of the Specific-region VAE and proposed to optimize both the Specific-region VAE and the De-drift Layer alternatively using trajectories in the target region.

Specifically, we denote the parameters in the Specific-region VAE as $\Theta = \{\phi'_e, \phi'_\mu, \phi'_\sigma, \phi'_d, \theta'\}$ and the parameters in the De-drift Layer as $\theta^\sigma = \{\theta^\sigma_s, \theta^\sigma_b\}$. In the Specific-region VAE optimization, we sample the trajectories to update the parameters in the Specific-region VAE while fastening the scaling and bias term. The operation can be denoted as:

$$\Theta = \Theta - \alpha \nabla_\Theta \mathcal{L}_{Spec} \tag{15}$$

where α is the learning rate. In the De-drift Layer optimization, we update the scaling and bias term while keeping the Specific-region VAE fixed. Particularly, we focus on the variables of the reconstruction loss $\mathcal{L}^s_{rec}$ between cases with and without the De-biasing Layer to evaluate its effectiveness and update the scaling and bias terms. The operation can be represented as:

$$\theta^\sigma = \theta^\sigma - \beta \nabla_{\theta^\sigma} |\mathcal{L}^{\text{VAE}_{Spec}+\text{Ded}}_{rec} - \mathcal{L}^{\text{VAE}_{Spec}}_{rec}| \tag{16}$$

where $\mathcal{L}^{\text{VAE}_{Spec}+\text{Ded}}_{rec}$ is the reconstruction loss calculated with the De-drift Layer and $\mathcal{L}^{\text{VAE}_{Spec}}_{rec}$ is without it. β is the learning rate.

C. OUTLIER DETECTION

With the process mentioned above, we have learned an encoder(denoted as Enc_{det}) to model the trajectories in the target region. Specifically, the Enc_{tar} consists of the Global VAE's encoder, the Specific-VAE's encoder and the De-drift Layer, which is able to map the trajectories to the target feature space and obtain an expressive representation $\mathbf{h}^{out}_T$. To detect outlier trajectories, we further add a classifier that follows the De-drift Layer to identify the outliers and optimized with cross entropy function. The operation can be denoted as:

$$\begin{aligned} \mathbf{h}^{out}_T &= \text{Enc}_{tar}(\mathbf{e}_{i...L}) \\ \mathbf{o}_T &= \text{softmax}(\text{MLP}(\mathbf{h}^{out}_T)) \\ \mathcal{L}_{tar} &= \text{CE}(\mathbf{o}_T, l_T) \end{aligned} \tag{17}$$

where $\mathbf{h}^{out}_T$ is the trajectory representation in the target feature space, $\mathbf{o}_T$ is the output of the classifier, and l_T is the label.

D. OPTIMIZATION

To optimize our model, we adopt a two-stage training manner that involves pre-training and fine-tuning. In the pre-training, we update the parameters of the Global VAE based on the loss function in Equation 8, which is computed using the unlabeled trajectories in the auxiliary regions. In the fine-tuning, we leverage the trajectories in the target region for optimization. Specifically, we first freeze the parameters of the Global VAE and then iteratively optimize the parameters of the Specific-region VAE and De-drift Layer using Equations 15 and 16 respectively. After this optimization, we proceed to train the classifier using Equation 17 to identify outliers. Specifically, to enhance the adaptation of the encoder Enc_{det} to the outlier detection task, we tie the parameter of the Specific-VAE's encoder and the classifier together for adjustment, while keeping the other components fixed.

V. EXPERIMENT

To assess the effectiveness of our proposed TTOD, we conducted experiments on two real trajectory datasets with the aim of addressing the following research questions (**Qs**):

- **Q1**: How effective is TTOD in detecting trajectory outliers in regions with sparse data?
- **Q2**: How do the main components contribute to the performance of TTOD?
- **Q3**: How does the selection of Gaussian distribution numbers in the latent space impact the performance of the model?
- **Q4**: How does the model's performance vary when fine-tuning with different number of trajectories in the target region?

A. EXPERIMENTAL SETTINGS

In this section, we introduce the datasets, evaluation metrics, compared baselines, and implementation details in our experiment. Subsequently, we present the results and compare them with the baseline methods.

1) DATASETS

To evaluate our approach, we utilize two taxi trajectory datasets collected in Xi'an and Chengdu by DiDi Inc. We additionally divide each city into 8 sub-regions for city-wide trajectory outlier detection and choose trajectories with a length greater than 20 in each region to construct training tasks. Then for each city, we select 5 regions as auxiliary regions and the other 3 as target regions. Since there were no outlier trajectories in the dataset, we generate outlier trajectories with a Dijkstra-based method and the details can be found in the following. After the above processing, The details about the processed datasets are shown in Table 2.

a: OUTLIER TRAJECTORY GENERATION

We first select a set of start and destination locations, and determine their normal routes on the Road Network. Then randomly remove one or more road segments from the Road Network to construct a new route topological structure. Finally, we use the Dijkstra algorithm to generate outlier trajectories between the selected start and destination locations on the new Road Network.

2) EVALUATION METRICS

We utilize Precision, Recall, and F1-score as evaluation metrics [40] to assess the performance of TTOD.

a: PRECISION

This is a metric calculated as the ratio of True Positive Samples to the total number of positively detected samples. True Positive refers to the samples that are correctly identified as positive by the model and are actually positive. False Positive samples are the ones that are wrongly identified as positive by the model but are actually negative.

b: RECALL

This is a metric that measures the proportion of True Positive samples relative to the total number of actual positive

TABLE 2. Statistics of the processed datasets.

	Xi'an							
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$
Normal	8000	8000	7000	5000	4500	3000	3500	4000
Outlier	800	800	700	500	450	300	350	400
	Chengdu							
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$
Normal	8000	8000	5000	7500	5000	3000	3000	2500
Outlier	800	800	500	750	500	300	300	250

TABLE 3. Performance comparison w.r.t. Recall, Precision and F1-score.

Model	Xi'an			Chengdu		
	Recall	Precision	F1-score	Recall	Precision	F1-score
ATD-RNN	0.54	0.581	0.586	0.519	0.623	0.566
Traj2vec	0.718	0.679	0.698	0.71	0.736	0.723
EncDec-AD	0.76	0.719	0.739	0.77	0.724	0.746
VSAE	0.783	0.75	0.767	0.789	0.772	0.781
GM-VSAE	0.835	0.8	0.817	0.819	0.834	0.827
DeepTEA	0.859	0.807	0.832	0.841	0.825	0.833
TSAD	0.738	0.712	0.725	0.739	0.716	0.727
ForenTrans	0.817	0.791	0.804	0.832	0.791	0.811
Siamese	0.73	0.638	0.681	0.743	0.685	0.713
TTOD	**0.938**	**0.913**	**0.925**	**0.906**	**0.889**	**0.897**

instances which contains both True Positive and False Negative samples. False Negative samples are those that the model identified as negative, but are actually positive in reality.

c: F1-SCORE

This is a metric that aggregates Precision and Recall measures using the harmonic mean, which allows for finding the optimal balance between the two metrics.

3) COMPARED METHODS

We compare our approach with 6 existing methods for trajectory outlier detection and 3 for outlier detection with sparse data. The specific details are represented in follows:

a: BASELINE METHODS FOR TRAJECTORY OUTLIER DETECTION

- **ATD-RNN** [16]: This is a supervised method. It models trajectories using RNN and employs a classifier to identify anomalous trajectories.
- **Traj2vec** [42]: This is a semi-supervised approach to learning trajectory representations. The method utilizes sliding windows to extract statistical features from trajectories and then leverages seq2seq models to learn their representations. We further utilize cluster algorithms for outlier detection.
- **EncDec-AD** [41]: This is a semi-supervised method that introduces an LSTM-based encoder-decoder to reconstruct sequence data. It is optimized by minimizing the reconstruction error of normal sequences, and the sequences that cannot be reconstructed well are identified as outliers.
- **GM-VSAE** [21]: This is a semi-supervised method based on VAE. The method uses Gaussian mixture distribution to describe the latent trajectory patterns and identifies outliers based on the probability of the trajectory being generated from normal patterns.
- **VSAE** [21]: This is a simplified version of GM-VSAE, where the distribution of latent routes follows a Gaussian distribution.
- **DeepTEA** [22]: This is a semi-supervised method that aims to detect time-dependent trajectory outliers. The approach uses a CNN to model road traffic conditions and utilizes a Gaussian mixture VAE to capture the latent patterns of trajectories. Similarly to the **GM-VSAE**, the outliers are identified based on the generated probabilities of the trajectories.

b: BASELINE METHODS FOR OUTLIER DETECTION WITH SPARSE DATA

- **TSAD** [43]: This is a supervised method based on transfer learning for outlier detection in time series data, which involves pretraining a model on a large-scale synthetic univariate time series dataset, and then fine-tuning its output layers' parameters on a smaller target dataset while keeping the other layers fixed. To adapt to the trajectory outlier detection, we utilize an LSTM-based

encoder to learn the trajectory embeddings, and subsequently use a classifier to detect outliers. Our optimization process follows the reference [43], whereby we pre-train the model with trajectories in the auxiliary regions and then fine-tune the parameters of the classifier on the dataset in the target region. Since the trajectories in auxiliary regions are all unlabeled, we use the embeddings learned by **Traj2vec** to obtain pseudo-labels with cluster algorithms, and use the trajectories with pseudo-labels to create the pre-training dataset.

- **ForenTrans** [44]: This is a semi-supervised method based on autoencoder for image manipulation detection. The method aims to learn a encoder to extract the information needed for decision-making in the source domain, and then applies the learned model to new target domains. To implement this approach, we leverage a model with the same architecture as **EncDec-AD** and pre-trained in the auxiliary regions. And then add a classifier after the encoder for outlier detection and fine-tune in the target region.
- **Siamese** [45]: This is a supervised method based on metric learning that uses a Siamese Network to identify outliers, classifying samples with higher similarity scores with the outliers than with the normal data as anomalous samples.

4) IMPLEMENTATION DETAILS

Unless stated otherwise, TTOD employs a transformer consisting of 4 layers and 4 attention heads in the Global VAE. For the Specific-region VAE, we leverage 2 layer MLP as the backbone of its encoder and decoder. Moreover, We tune the hyperparameters with grid search. Specifically, the learning rate $\alpha = 0.001$ and $\beta = 0.01$. And the number of Gaussian components is 6. For the road network, we directly used the downtown areas of the two cities from OpenStreetMap [47].

B. EXPERIMENTAL RESULTS

We conducted an experiment for the performance comparison of our TTOD with the 8 baselines mentioned above, and the results are presented in Table 3. The results reveal various insights that can be interpreted in multiple ways and answer **Q1**. Our model shows significant advantages compared to both trajectory outlier detection methods and outlier detection methods with sparse data. Specifically, compared to the best baseline **DeepTEA**, on the Chengdu dataset, our model improved recall by 7%, precision by 6% and F1-score by nearly 6%. On the Xi'an dataset, the advantages are even more significant, with a 8% improvement in recall, a 11% improvement in precision and a 9% increase in F1-score.

In the baselines of trajectory outlier detection, **DeepTEA** performs the best while **ATD-RNN** performs the worst. The superior performance of **DeepTEA** can be attributed to its consideration of traffic conditions, which is achieved by the CNN. Additionally, using Gaussian mixture VAE to model the latent feature space of trajectories can also capture complex feature patterns. Moreover, the superiority of **GM-VSAE** over **VSAE** provides further evidence of the crucial role played by modeling the latent space with Gaussian mixture distributions. The **ATD-RNN** is a supervised method and requires a large amount of labeled trajectory data with balanced label distribution for model training. Therefore it performs poorly when there are limited labeled trajectories in the training stage.

ForenTrans outperforms all other baselines for outlier detection with sparse data, delivering superior and more consistent results. This method employs an autoencoder to model the feature space for knowledge transfer and is trained in a weakly-supervised learning manner. While it does not depend on large quantities of labeled data, it assumes that the feature space of the auxiliary regions and target region can be shared, which falls short in scenarios where feature distribution drift occurs. Furthermore, sharing the feature space for knowledge transfer in a straightforward manner creates an overly large search space for the target region, which can lead to unstable detection performance. As a popular method in few-shot learning studies, the **Siamese** is highly sensitive to the selection of typical normal and typical outlier samples. It can achieve good performance under some anchor samples. However, this method is extremely unstable and performs poorly when typical samples cannot be guaranteed to be existing or accurately selected.

C. ABLATION STUDY

To answer question **Q2**, we performed an ablation study to access the effectiveness of the main components in TTOD. The ablated variants are represented as follows:

- **TTOD** $_{w/o\ EmbVAE}$: This is a variant of TTOD with a single global VAE for feature space learning and optimized as **ForenTrans**.
- **TTOD** $_{w/o\ Dedrift}$: This is the degenerate model of TTOD with no De-drift Layer.

1) CONTRIBUTION OF THE NESTED STRUCTURE OF VAE

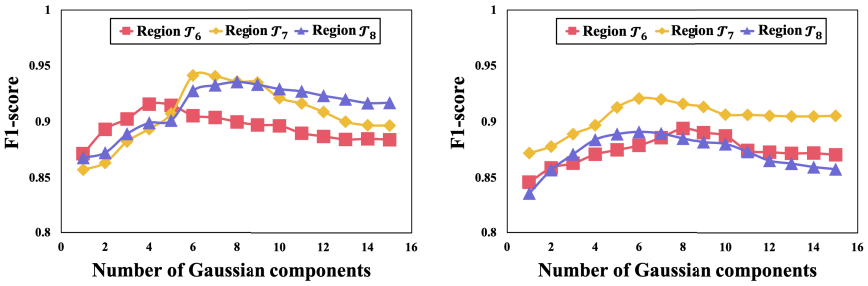
We compare the performance of our model and the ablated version without the Specific-region VAE and the De-drift Layer, *i.e.* TTOD $_{w/o,EmbVAE}$, to verify the effectiveness of the nested structure for knowledge transfer between the feature spaces of the auxiliary and target regions. As shown in Table 4, TTOD improved about 8% in F1-score, which can be interpreted in two ways. Firstly, there is a gap between the feature space in the auxiliary regions and the target regions. To facilitate knowledge transfer, a nonlinear transformation function is required. Secondly, the nested VAE architecture can effectively learn the mapping function between the feature spaces in the auxiliary and target regions, further enhancing the model's knowledge in the target region and improving its performance.

2) CONTRIBUTION OF DE-DRIFT LAYER

To validate the effectiveness of the De-drift layer, We compare the performance of TTOD and TTOD $_{w/o\ Dedrift}$.

TABLE 4. Performance comparison w.r.t. Recall, Precision and F1-score in ablation study.

	Xi'an			Chengdu		
	Recall	Precision	F1-score	Recall	Precision	F1-score
TTOD $_{w/o\ EmbVAE}$	0.839	0.85	0.845	0.824	0.836	0.83
TTOD $_{w/o\ Dedrift}$	0.879	0.846	0.862	0.853	0.877	0.865
TTOD	**0.938**	**0.913**	**0.925**	**0.906**	**0.889**	**0.897**



(a) The F1-score on Xi'an dataset

(b) The F1-score on Chengdu dataset

FIGURE 7. Varying the number of Gaussian components of TTOD.

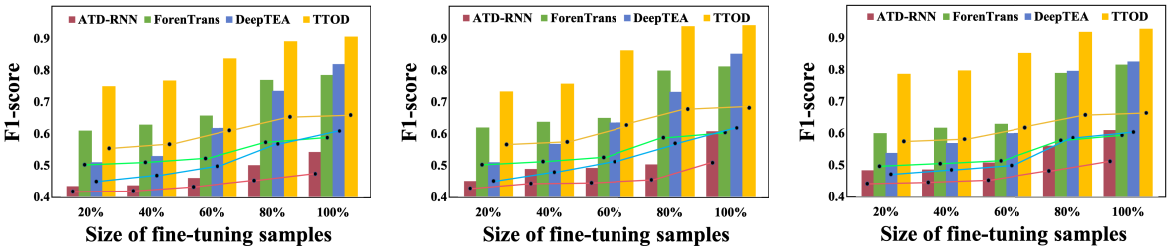
The results in Table 4 demonstrate the significant role of the De-drift layer in the outlier detection task in the target region. Specifically, the F1-score improved about 6% on the Xi'an dataset and about 3% on the Chengdu dataset. We attribute this to the transformation achieved by the De-drift Layer, which increases the diversity of the feature space by adding scale and bias that sampled from Gaussian distributions and further reduces or even eliminates the unseen-subspace of the target feature space mentioned in Section I.

D. SENSITIVITY ANALYSIS

Since our model employs Gaussian mixture distributions to construct the feature space, the number of Gaussian components directly affects the model's ability to model trajectories. In this section, we evaluate the model's performance under various numbers of Gaussian components to answer question **Q3**. As shown in Figure 7, the performance of the model fluctuates significantly as the value of C changes and exhibits similar trends in all the target regions. Specifically, when the C changes from 1 to 10, the performance shows an upward trend, followed by a downward trend, and then gradually stabilizes. It is easy to understand that when $C = 1$, the model degenerates into a traditional VAE model with limited expressive ability in the feature space, resulting in poor performance. As C increases, multiple Gaussian distributions can describe more complex trajectory features, thereby enhancing its ability to model trajectories and significantly improving performance. As C increases beyond the optimal value, redundant Gaussian components introduce additional noise into the feature space, leading to a decline in performance. For example, in the region $\mathcal{T}_7$ of Xi'an, when the number of Gaussian components is 6, the model's performance reaches its optimum, indicating that a mixture of 6 Gaussian distributions is sufficient to model the feature space of the trajectories in the region. After exceeding 6, the performance shows a downward trend.

E. PERFORMANCE ANALYSIS OF THE MODEL WITH DIFFERENT NUMBER OF FINE-TUNING TRAJECTORIES IN THE TARGET REGION

To answer question **Q4**, we compare TTOD with three typical baseline methods (*i.e.* **ATD-RNN**, **DeepTEA** and **ForenTrans**) on the Xi'an dataset to analyze the performance of the model with the different number of fine-tuning trajectories. Specifically, we vary the size of the dataset for fine-tuning from 20% to 100% proportions of the target regions' training dataset, and then evaluated the model's performance on the test dataset. As shown in Figure 8, TTOD achieves better performance in all experiment settings and demonstrated excellent stability. Taking the results on Region $\mathcal{T}_7$ as an example, when the proportion of fine-tuning samples is above 60%, the model's performance remains relatively stable despite the decrease in data size, highlighting its robustness in handling changes in sample quantity. As the proportion falls below 60%, there is a noticeable decline in performance initially, but it quickly stabilizes. This can be easily understood as the performance decline is primarily attributed to the reduction in the inherent information content of the smaller dataset. Furthermore, even in the extreme case where only 20% of the dataset is available, the model consistently maintains an F1-measure of over 70%, showcasing its superiority in handling sparse data scenarios. Among all the baseline methods, **ForenTrans** demonstrates better stability, primarily due to the knowledge acquired during the pre-training phase, which effectively enhances detection in target regions. However, its performance is significantly inferior to that of TTOD. **DeepTEA** and **ATD-RNN** exhibit notable declines in performance as the sample quantity decreased. This can be



(a) The F1-score in region $\mathcal{T}_6$ (b) The F1-score in region $\mathcal{T}_7$ (c) The F1-score in region $\mathcal{T}_8$

FIGURE 8. The performance comparison of TTOD and the selected baselines w.r.t. different sizes of the fine-tuning samples on Xi'an dataset.

attributed to the fact that **DeepTEA** requires a substantial amount of training samples to learn latent patterns of normal trajectories. Similarly, **ATD-RNN** also relies on a significant labeled dataset to acquire effective features for detection.

VI. CONCLUSION

In this paper, we focus on the region-wide trajectory outlier detection with sparse data. Considering the challenges in data scarcity and the data drift problem among regions. We proposed a VAE-based model that is designed to transfer the knowledge in feature space from the auxiliary regions to the target region. Specifically, to learn the Global feature space with the trajectories in auxiliary regions, we introduced the Global VAE which is a Transformer-based model and describes the feature space with Gaussian mixture distributions. To transfer the learned knowledge to the target region, we then leverage an MLP-based VAE, *i.e.* the Specific-region VAE, to map the learned global feature space to the target feature space. As for the data drift problem among regions, we introduced the De-drift Layer to generate novel feature patterns and approximate the unseen feature space. Extensive experiments on two trajectory datasets demonstrate that the proposed method significantly outperforms all the compared baselines.

REFERENCES

[1] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong, "A survey on trajectory data management, analytics, and learning," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2022.
[2] Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "Trajectory outlier detection: New problems and solutions for smart cities," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 2, pp. 1–28, Apr. 2021.
[3] Z. Zhu, D. Yao, J. Huang, H. Li, and J. Bi, "Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2018, pp. 1–12.
[4] D. Zhang, Z. Chang, S. Wu, Y. Yuan, K.-L. Tan, and G. Chen, "Continuous trajectory similarity search for online outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4690–4704, Oct. 2022.
[5] Z. Lv, J. Xu, P. Zhao, G. Liu, L. Zhao, and X. Zhou, "Outlier trajectory detection: A trajectory analytics based approach," in *Database Systems for Advanced Applications*. Cham, Switzerland: Springer, Mar. 2017, pp. 231–246.
[6] J. Lee, G. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 140–149.
[7] E. Masciari, "Trajectory outlier detection using an analytical approach," in *Proc. IEEE 23rd Int. Conf. Tools Artif. Intell.*, Nov. 2011, pp. 377–384.
[8] L. Liu, S. Qiao, Y. Zhang, and J. Hu, "An efficient outlying trajectories mining approach based on relative distance," *Int. J. Geographical Inf. Sci.*, vol. 26, no. 10, pp. 1789–1810, Oct. 2012.
[9] M. A. Saleem, W. Nawaz, Y.-K. Lee, and S. Lee, "Road segment partitioning towards anomalous trajectory detection for surveillance applications," in *Proc. IEEE 14th Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2013, pp. 610–617.
[10] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Effective and efficient trajectory outlier detection based on time-dependent popular route," *World Wide Web*, vol. 20, no. 1, pp. 111–134, Jan. 2017.
[11] Y. Wang, K. Qin, Y. Chen, and P. Zhao, "Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 1, p. 25, Jan. 2018.
[12] X. Ying, Z. Xu, and W. G. Yin, "Cluster-based congestion outlier detection method on trajectory data," in *Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2009, pp. 1–11.
[13] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, "Detecting moving object outliers in massive-scale trajectory streams," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 422–431.
[14] T. Zhang, S. Zhao, and J. Chen, "Ship trajectory outlier detection service system based on collaborative computing," in *Proc. IEEE World Congr. Services*, 2018, pp. 15–16.
[15] Y. Huang and Q. Zhang, "Identification of anomaly behavior of ships based on KNN and LOF combination algorithm," in *Proc. AIP Conf.*, 2019, vol. 2073, no. 1.
[16] L. Song, R. Wang, D. Xiao, X. Han, Y. Cai, and C. Shi, "Anomalous trajectory detection using recurrent neural network," in *Proc. Int. Conf. Adv. Data Mining Appl.* Cham, Switzerland: Springer, Nov. 2018, pp. 263–277.
[17] R. R. Sillito and R. B. Fisher, "Semi-supervised learning for anomalous trajectory detection," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, 2008, p. 35-1.
[18] Y. Cheng, B. Wu, L. Song, and C. Shi, "Spatial–temporal recurrent neural network for anomalous trajectories detection," in *Proc. Int. Conf. Adv. Data Mining Appl.* Cham, Switzerland: Springer, Nov. 2019, pp. 565–578.
[19] K. Gray, D. Smolyak, S. Badirli, and G. Mohler, "Coupled IGMM-GANs for improved generative adversarial anomaly detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2538–2541.
[20] H. Wu, W. Sun, and B. Zheng, "A fast trajectory outlier detection approach via driving behavior modeling," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1–13.
[21] Y. Liu, K. Zhao, G. Cong, and Z. Bao, "Online anomalous trajectory detection with deep generative sequence modeling," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 949–960.
[22] X. Han, R. Cheng, C. Ma, and T. Grubenmann, "DeepTEA: Effective and efficient online time-dependent trajectory outlier detection," *Proc. VLDB Endowment*, vol. 15, no. 7, pp. 1493–1505, Mar. 2022.
[23] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "IBAT: Detecting anomalous taxi trajectories from GPS traces," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, Sep. 2011, pp. 99–108.
[24] J. Huang, M. Deng, J. Tang, S. Hu, H. Liu, S. Wariyo, and J. He, "Automatic generation of road maps from low quality GPS trajectory data via structure learning," *IEEE Access*, vol. 6, pp. 71965–71975, 2018.
[25] T. Zou, J. Angeles, and F. Hassani, "Dynamic modeling and trajectory tracking control of unmanned tracked vehicles," *Robot. Auto. Syst.*, vol. 110, pp. 102–111, Dec. 2018.
[26] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.
[27] Q. Yu, Y. Luo, C. Chen, and X. Wang, "Trajectory outlier detection approach based on common slices sub-sequence," *Int. J. Speech Technol.*, vol. 48, no. 9, pp. 2661–2680, Sep. 2018.

[28] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4362–4369, Jul. 2019.
[29] P. Yang, D. Wang, Z. Wei, X. Du, and T. Li, "An outlier detection approach based on improved self-organizing feature map clustering algorithm," *IEEE Access*, vol. 7, pp. 115914–115925, 2019.
[30] A. Smiti, "A critical overview of outlier detection methods," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100306.
[31] A. Carreño, I. Inza, and J. A. Lozano, "Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework," *Artif. Intell. Rev.*, vol. 53, no. 5, pp. 3575–3594, Jun. 2020.
[32] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," *Big Data Cognit. Comput.*, vol. 5, no. 1, p. 1, Dec. 2020.
[33] Z. Wang, G. Yuan, H. Pei, Y. Zhang, and X. Liu, "Unsupervised learning trajectory anomaly detection algorithm based on deep representation," *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 12, Dec. 2020, Art. no. 155014772097150.
[34] J. Zhao, Z. Yi, S. Pan, Y. Zhao, Z. Zhao, F. Su, and B. Zhuang, "Unsupervised traffic anomaly detection using trajectories," in *Proc. CVPR Workshops*, Jun. 2019, pp. 133–140.
[35] Y. Yao, X. Wang, M. Xu, Z. Pu, Y. Wang, E. Atkins, and D. J. Crandall, "DoTA: Unsupervised detection of traffic anomaly in driving videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 444–459, Jan. 2023.
[36] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, and G. Fortino, "Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection," *Inf. Fusion*, vol. 65, pp. 13–20, Jan. 2021.
[37] A. Belhadi, Y. Djenouri, J. C.-W. Lin, and A. Cano, "Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges," *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 3, pp. 1–29, Sep. 2020.
[38] L. Shi, C. Huang, M. Liu, J. Yan, T. Jiang, Z. Tan, Y. Hu, W. Chen, and X. Zhang, "UrbanMotion: Visual analysis of metropolitan-scale sparse trajectories," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 10, pp. 3881–3899, Oct. 2021.
[39] L. ZhanLi and Y. JiaWei, "Abnormal behavior recognition based on transfer learning," *J. Phys., Conf. Ser.*, vol. 1213, no. 2, Jun. 2019, Art. no. 022007.
[40] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," 2020, *arXiv:2008.05756*.
[41] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder–decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*.
[42] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3880–3887.
[43] T. Wen and R. Keyes, "Time series anomaly detection using convolutional neural networks and transfer learning," 2019, *arXiv:1905.13628*.
[44] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," 2018, *arXiv:1812.02510*.
[45] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5790–5798, Aug. 2021.
[46] W. Dong, D. Yan, and P. Wang, "Self-supervised node representation learning via node-to-neighbourhood alignment," 2023, *arXiv:2302.04626*.
[47] J. Bennett, *OpenStreetMap*. Birmingham, U.K.: Packt Publishing, 2010.
[48] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.
[49] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–16.
[51] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[52] D. Yao, G. Cong, C. Zhang, and J. Bi, "Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1358–1369.
[53] Q. Jing, D. Yao, C. Gong, X. Fan, B. Wang, H. Tan, and J. Bi, "TrajCross: Trajecotry cross-modal retrieval with contrastive learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 344–349.
[54] D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 2411–2414.
[55] D. Yao, C. Zhang, Z. Zhu, Q. Hu, Z. Wang, J. Huang, and J. Bi, "Learning deep representation for trajectory clustering," *Expert Syst.*, vol. 35, no. 2, Apr. 2018, Art. no. e12252.
[56] D. Yao, G. Cong, C. Zhang, X. Meng, R. Duan, and J. Bi, "A linear time approach to computing time series similarity based on deep metric learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4554–4571, Oct. 2022.
[57] W. Li, X. Chu, Y. Su, D. Yao, S. Zhao, R. Wu, S. Zhang, J. Tao, H. Deng, and J. Bi, "FingFormer: Contrastive graph-based finger operation transformer for unsupervised mobile game bot detection," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 3367–3375.
[58] Y. Su, D. Yao, X. Chu, W. Li, J. Bi, S. Zhao, R. Wu, S. Zhang, J. Tao, and H. Deng, "Few-shot learning for trajectory-based mobile game cheating detection," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 3941–3949.

YUEYANG SU (Student Member, IEEE) is currently pursuing the Ph.D. degree in computer architecture with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include data mining, machine learning, and time series anomaly detection.

DI YAO (Member, IEEE) received the Ph.D. degree from the University of Chinese Academy of Sciences under the supervision of Prof. Jingping Bi. He is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include spatio-temporal data mining and deep learning.

TIAN TIAN (Member, IEEE) received the Ph.D. degree from Southeast University. He is currently a Senior Engineer with the Nanjing Marine Radar Institute, specializing in the research of radar information processing technologies, particularly in the areas of electronic data mining and machine learning for information processing.

JINGPING BI (Member, IEEE) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2002. She is currently a Full Professor with the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include network measurement, routing, virtualization, SDN, and big data.