Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A privacy-preserving vehicle trajectory clustering framework*

Ran TIAN‡, Pulun GAO, Yanxing LIU

College of Computer Science & Engineering, Northwest Normal University, Lanzhou 730070, China

E-mail: tianran@nwnu.edu.cn; 202031603111@nwnu.edu.cn; lyanxing@nwnu.edu.cn

Received May 23, 2023; Revision accepted July 21, 2023; Crosschecked June 21, 2024

Abstract: As one of the essential tools for spatio–temporal traffic data mining, vehicle trajectory clustering is widely used to mine the behavior patterns of vehicles. However, uploading original vehicle trajectory data to the server and clustering carry the risk of privacy leakage. Therefore, one of the current challenges is determining how to perform vehicle trajectory clustering while protecting user privacy. We propose a privacy-preserving vehicle trajectory clustering framework and construct a vehicle trajectory clustering model (IKV) based on the variational autoencoder (VAE) and an improved *K*-means algorithm. In the framework, the client calculates the hidden variables of the vehicle trajectory and uploads the variables to the server; the server uses the hidden variables for clustering analysis and delivers the analysis results to the client. The IKV' workflow is as follows: first, we train the VAE with historical vehicle trajectory data (when VAE's decoder can approximate the original data, the encoder is deployed to the edge computing device); second, the edge device transmits the hidden variables to the server; finally, clustering is performed using improved *K*-means, which prevents the leakage of the vehicle trajectory. IKV is compared to numerous clustering methods on three datasets. In the nine performance comparison experiments, IKV achieves optimal or sub-optimal performance in six of the experiments. Furthermore, in the nine sensitivity analysis experiments, IKV not only demonstrates significant stability in seven experiments but also shows good robustness to hyperparameter variations. These results validate that the framework proposed in this paper is not only suitable for privacy-conscious production environments, such as carpooling tasks, but also adapts to clustering tasks of different magnitudes due to the low sensitivity to the number of cluster centers.

Key words: Privacy protection; Variational autoencoder; Improved *K*-means; Vehicle trajectory clustering
https://doi.org/10.1631/FITEE.2300369 **CLC number:** TP183

1 Introduction

With the development of location acquisition and mobile computing technology, we can obtain a large amount of trajectory data from mobile objects (Zheng, 2015). Using specific information mining methods, we can extract relevant knowledge. For example, clustering analysis of ship trajectory data can estimate ship arrival time (Xu et al., 2022). Mining typhoon trajectories can provide early warning of flooding (Chang et al., 2020). The analysis of multi-dimensional trajectory data allows online monitoring of abnormal behaviors (Pan et al., 2020). In this paper we focus on vehicle trajectories. Vehicle trajectory clustering is a suitable tool that can combine similar vehicle trajectories into a single cluster, which is widely used for tasks such as travel pattern detection (Hong et al., 2018), early driver intention prediction (Yi et al., 2019), path choice set identification (Advani et al., 2022), and ride matching for commuting trips (Hong et al., 2017).

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 71961028), the Key Research and Development Program of Gansu Province, China (No. 22YF7GA171), the University Industry Support Program of Gansu Province, China (No. 2023QB-115), the Innovation Fund for Science and Technology-Based Small and Medium Enterprises of Gansu Province, China (No. 23CXGA0136), the Traditional Chinese Medicine Industry Innovation Consortium Project of Gansu Province, China (No. 22ZD6FA021-5), and the Scientific Research Project of the Lanzhou Science and Technology Program, China (No. 2018-01-58)

ORCID: Ran TIAN, https://orcid.org/0000-0003-4435-580X; Pulun GAO, https://orcid.org/0009-0001-8889-8227; Yanxing LIU, https://orcid.org/0000-0002-0554-3683

Vehicle trajectory clustering can be categorized into two types, i.e., without or with privacy protection. Research on vehicle trajectory clustering without privacy protection focuses on how to improve the clustering quality of the respective studied roadway or application scenarios (Yu et al., 2019; Chen et al., 2021; Wang W et al., 2021). However, personal information, such as user hobbies, eating habits, and commuting patterns, is easily stolen using trajectory clustering and other methods. The methods of uploading original vehicle trajectory data directly to the server may leak sensitive personal information and cause irreparable harm to users. Research on vehicle trajectory clustering with privacy protection focuses on how to achieve vehicle trajectory clustering while protecting user privacy. The representative studies are *K*-anonymity (Oksanen et al., 2015), differential privacy (Wang H and Xu, 2018), and federated learning (Kong and Lu, 2022). However, these privacy-preserving techniques may lessen clustering effects if protecting privacy.

Therefore, one of this study's challenges is to achieve privacy protection while still ensuring the usefulness of the trajectory encodings uploaded to the cloud for data analysis. In other words, privacy protection requires the balancing of two metrics: privacy and utility (Jin et al., 2023). To address this challenge, we have made the following contributions to this research:

1. We propose a privacy-preserving vehicle trajectory clustering framework, which generates the hidden variables of the original data locally and uploads the hidden variables to the server for clustering. On one hand, the original private data on vehicle trajectory are safeguarded. On the other hand, the hidden variables can significantly improve the clustering effect.

2. We propose a vehicle trajectory clustering model (IKV) based on the variational autoencoder (VAE) and an improved *K*-means algorithm. The model uses historical trajectory data of vehicles for training on the VAE and calculating latent variables, which helps discover the intrinsic patterns in the trajectory data while protecting privacy, thereby improving the effect of clustering.

3. We have improved the *K*-means algorithm. In each iteration, the cluster centers are updated by minimizing the error value from the cluster center to other trajectory points. The trajectory objects that match the cluster centers screened by the IKV can be found in the real world.

4. Experiments on three real-world datasets reveal that the IKV model outperforms other comparative methods on most evaluation metrics. Moreover, in sensitivity analysis, IKV shows good stability and robustness.

2 Related works

The most common clustering algorithms operate on data that are made up of discrete points with specific dimensions. On the other hand, the vehicle trajectory data comprise time series that have variable length and inconsistent dimensionality. For the characteristics of vehicle trajectory data changing on the axis of time, Gaffney and Smyth (1999) treated vehicle trajectory as a function whose dependent variable is the timestamp, and performed trajectory clustering. For the characteristics of the inconsistent dimension of vehicle trajectory data, research has been conducted on the setting of time domain and time interval (Benkert et al., 2008; Atev et al., 2010). The above works provided the foundation for sampling vehicle trajectory data by time domain and time interval. Using the trajectory continuity feature, Gariel et al. (2011) re-sampled the trajectories to generate new datasets, and then subjected these time series to principal component analysis (PCA) before using them as embedding vectors in the clustering algorithm. The PCA improves the clustering efficiency, but in this study we conduct extensive experiments on various dimensionality reduction algorithms and discover that VAE is more suitable for the clustering of road networks than PCA. Furthermore, we adjust the strategy's structure to deploy the dimensionality reduction component to the client, preventing the server from obtaining the user data.

Gariel et al. (2011)'s strategy works well on airplane routes but is inapplicable to complicated and varied road networks. Chen et al. (2021) considered road connection, road direction, the actual road length, and other road network-sensitive factors for clustering and limited the clustering using the real circumstances of the actual road section's characteristics. Chen et al. (2021)'s strategy addresses the influence of sensitive parameters on the clustering effect in the specific

segment. However, the artificially set constraints are based on extensive knowledge of the studied segment, which makes implementation more difficult and increases storage costs and computing costs. As for our proposed algorithm, the input data need only to contain the coordinates of the road segment, so it can be applied to various road sections. To address the problem of the diversity of driving patterns in road networks, Yu et al. (2019) proposed a multi-feature-based method for estimating distances and clustering of vehicle trajectories based on driving speed, direction, form, continuity, and other factors. Therefore, Yu et al. (2019)'s strategy requires a dataset with specific properties. As for the data of our proposed algorithm, the features describing driving patterns do not have to be specific attributes, so our proposed system can handle a wide range of datasets. To avoid the problems caused by setting constraints, Besse et al. (2016) proposed a novel distance to describe the similarity between trajectory objects that does not rely on prior knowledge and is entirely data-driven. To address the issue of cost, Wang W et al. (2021) used the silhouette coefficient (Rousseeuw, 1987), the Davies–Bouldin index (Davies and Bouldin, 1979), and the Calinski–Harabaz index (Caliński and Harabasz, 1974) as metrics for evaluating clustering results, which simplifies the evaluation process and makes the evaluation of clustering results more efficient. The algorithms above provide solutions for applying clustering algorithms in different scenarios. However, sensitive information, such as user hobbies, eating habits, and commuting patterns, can be mined in vehicle tracking data. If the clustering is done directly using the original user's vehicle data, the sensitive information will possibly be exposed.

Using k-anonymous processing on the data is a viable solution to the privacy protection problem of trajectory information (Guo et al., 2015; Oksanen et al., 2015). Each trajectory is clustered with at least $k-1$ other trajectories to form an anonymous set, which is subsequently uploaded to the cloud. The attacker has only a tiny probability of capturing the correct trajectory, thus achieving privacy protection. The k-anonymous approach is static and samples only specific time segments, resulting in a significant loss of information from the vehicle trajectory data. Xin et al. (2017) proposed an adaptive dynamic trajectory publication algorithm that addresses the issue of inconsistency in sampling time and movement speed. Although the original user data were posted anonymously, they were still directly shared, so privacy protection was restricted. Furthermore, Gazdag et al. (2023) showed that anonymization techniques can provide only erratic, empirical privacy protection for a whole population, but not worst-case privacy protection for each individual. Therefore, they promoted principled differential privacy to anonymize data. Based on differential privacy, researchers have made different contributions by combining their concerns. Wang H and Xu (2018) proposed a general differential privacy-preserving method that applies to most trajectory clustering algorithms. Ma et al. (2019) proposed a privacy-preserving technology with differential privacy, called RPTR, which can provide better protection for areas with high user density. Arif et al. (2021) combined differential privacy with a generalized anonymization method, protecting the privacy of sensitive vehicle trajectories without destroying the spatio–temporal integrity of vehicle trajectories. Zhao et al. (2020) considered the uncertainty of the dimensional correlation amongst trajectory data and limited the size of the added noise, resulting in a method that successfully reduces noise and has good clustering performance. Kong and Lu (2022) used a Gaussian differential privacy mechanism to encrypt the decoder to prevent the attacker from using the decoder to restore the original information after the decoder information was leaked.

Both anonymization approaches and differential privacy approaches require uploading data to a trusted server. These approaches are too dependent on a single cloud server. Moreover, the computation is concentrated on the server, which has a high computational overhead. Xie et al. (2023) proposed the blockchain-based efficient privacy-preserving handover authentication protocol (BEPHAP), which solves the problem of the authentication process relying on a single cloud server and has a high computational overhead. Another decentralized technology is federated learning, which is also employed in vehicle trajectory clustering (Kong and Lu, 2022; Lu et al., 2023). During federated learning model training, only the model weights are uploaded to the server locally, while personal vehicle trajectories are not uploaded. Although the above method can avoid the direct uploading of raw data and ensure that the data change without significantly altering the

clustering result, it has difficulty increasing clustering quality and has some weaknesses. Our proposed framework's hidden variables can prevent the server from obtaining the driver's current vehicle trajectory data and increase clustering quality to a greater extent.

Another privacy-preserving strategy is to use the encoder to learn the low-dimensional representation of data (Cho et al., 2014), which compresses features into a one-dimensional feature space and obtains latent space features that only machines can comprehend. Performing clustering using the one-dimensional features avoids the direct sharing of users' sensitive data. Furthermore, because the encoder learns the main properties of the data, it can improve the clustering quality. For this approach, we conduct extensive tests using various algorithms and finally conclude that the quality of the hidden variables generated by VAE is relatively high.

3 Privacy-preserving vehicle trajectory clustering framework

3.1 System framework

To perform vehicle trajectory clustering while protecting user privacy and maximizing clustering effects, we propose a framework as shown in Fig. 1.

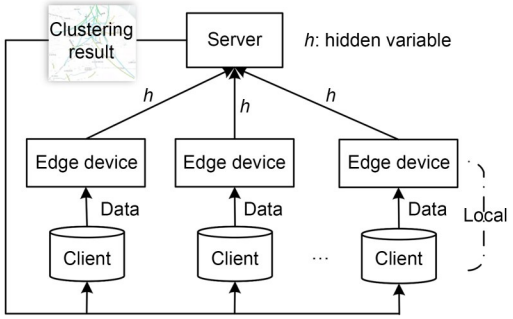

Fig. 1 Privacy-preserving vehicle trajectory clustering framework

Fig. 1 shows that the framework has two vital parts: client and server. (1) The client calculates the hidden variables of the original vehicle trajectory using a local edge computing device and uploads the hidden variables to the server. (2) The server performs vehicle trajectory clustering using hidden variables of vehicle trajectory data and distributes the clustering results to the individual clients. The original vehicle trajectory remains local throughout the framework and the attacker cannot infer the original data from the hidden variables uploaded to the server, ensuring user data privacy.

The framework models, deployed by the client and server, are replaceable. (1) In the client side, to avoid using the original vehicle trajectory data, the edge devices need to compute the low-dimensional representation of the vehicle trajectory data in advance. Therefore, the edge devices can use dimensionality reduction algorithms, such as PCA, autoencoder (AE), and singular value decomposition (SVD). (2) In the server side, to classify vehicle trajectories into different clusters, the model deployed on the server can choose the clustering algorithm, such as *K*-means, *K*-medoids, and fuzzy c-means (FCM).

3.2 Problem definition

3.2.1 The models deployed on the client

Definition 1 (Vehicle trajectory data) Define a set of vehicle trajectories with n vehicles $V=\{v_1, v_2, \cdots, v_n\}$, where a vehicle's trajectory is defined as the trajectory points on a dynamically changeable time slice. Therefore, for any vehicle object v_i, its sequence can be represented as $v_i=(p_{i1}, p_{i2}, \cdots, p_{il})$, where l is of indefinite length, and p_{ij} is the j^{th} trajectory point for the i^{th} vehicle. A trajectory point has a timestamp, longitude, latitude, and other trajectory-related attributes, denoted as $p=(t, \text{lon}, \text{lat}, \text{attr})$, and p contains a huge quantity of user-sensitive information.

Definition 2 (Self-encoding of sequences) Define a self-encoder AE, which includes an encoder AE_e and a decoder AE_d. AE_e needs to learn a low-dimensional representation z of a sequence x that can only be understood by a machine, so that z contains as much information as possible about x, and the decoder restores z to approximate x as closely as possible. The procedure can be denoted as

$$\min L(x, \text{AE}_\text{d}(\text{AE}_\text{e}(x))), \tag{1}$$

where $L(\cdot,\cdot)$ denotes the distance.

Definition 3 (Hidden variables) According to Definition 1, we assume a dataset with a set of vehicles $V=\{v_1, v_2, \cdots, v_n\}$ ($V\in\mathbb{R}^n$). The original set of trajectories is sliced to form a unified dimension, which is then

inputted to AE_e in Definition 2 to obtain the hidden variable $z \in \mathbb{R}^{n \times d}$, which maps the original data to the latent space, and d is the dimension of the latent space. When Eq. (1) satisfies certain expectations, the trajectory reverted through AE_d is quite similar to that of V. Any trajectory object v_i in the original data can be represented by the corresponding hidden variable z_i.

Based on the above definitions, the goal of this client is as follows: the client needs to train an AE using historical trajectory data. When the AE's decoder can provide an approximation to the original data, the encoder part of the AE will be deployed on the edge computing device of the client. Then, the edge computing device computes the set of hidden variables for the vehicle trajectory data and uploads the hidden variables to the server, which avoids directly using the original user's sensitive data.

For models deployed on the edge devices, we experiment with a large number of models. When compared to the baselines, using VAE to produce hidden variables performs better. Therefore, the client of this framework will deploy VAE.

3.2.2 The models deployed on the server

Definition 4 (Clusters) We define a set of vehicles $V=\{v_1, v_2, \cdots, v_n\}$ and a set $T=\{t_1, t_2, \cdots, t_{n'}\}$. When $T \subseteq V$, T is a cluster for V.

To find vehicles with similar behavioral patterns, the server needs to divide the vehicle set V into multiple clusters. The cluster set can be denoted as $C=\{T_1, T_2, \cdots, T_K\}$, where K is the number of clusters, and $\forall T_i \in C, \forall T_j \in C, T_i \cap T_j = \varnothing (i \neq j)$. Therefore, the goal of this server is to compute the clustering result using current vehicle trajectory data.

For models deployed on the server, the framework uses the division-based clustering algorithm.

3.3 System model

To achieve the objectives mentioned in Section 3.2, we assemble the better-performing VAE and improved K-means (IK) to the frame to obtain the IKV, whose overall process is shown in Fig. 2.

Fig. 2 shows that IKV is divided into three major phases. First, in the server side, the VAE is trained using historical vehicle trajectories. When the target of $\min L(X, X')$ fulfills specific expectations, the VAE's encoder is deployed to the vehicle's edge computing device. Second, after the edge computing device compresses the original vehicle trajectory into the latent space hidden variables, the local edge computing device transmits the hidden variables to the server. Finally, the clustering analysis is performed on the server side using the hidden variables and the improved K-means. The exact computation process is as follows:

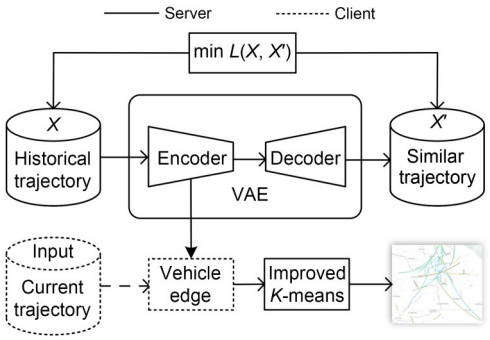

Fig. 2 Overall process of IKV

3.3.1 Training VAE using historical trajectories

Sample historical trajectory data by time domain and time interval to obtain consistent-length vehicle trajectory data X ($X \in \mathbb{R}^{N \times n}$). After putting X into the encoder, the hidden variable z ($z \in \mathbb{R}^{N \times m}$) is computed, where n and m are the feature dimensions of X and z respectively, and N represents the batch size. Because the gap between n and m is too high, we transmit X across three fully connected layers, gradually decreasing the latitude. The calculating equation is as follows:

$$h^k = \sigma(\mathrm{fc}_3(\sigma(\mathrm{fc}_2(\sigma(\mathrm{fc}_1(X^k)))))), \tag{2}$$

where $X^k \in \mathbb{R}^{N \times n}$ is the input data, fc_1, fc_2, and fc_3 are three fully connected layers, $\sigma(\cdot)$ is the rectified linear unit (ReLU) activation function, h^k is the final output of the hidden layer, and k is the batch number. The VAE independently fits the distribution of each input batch.

To obtain the distribution of X^k, h^k is sent in parallel to the two fully connected layers fc_4 and fc_5, which fit the mean and the squared difference of X^k simultaneously. The calculating equation is as follows:

$$N(\mu, \ln\sigma^2) = N(\mathrm{fc}_4(h^k), \mathrm{fc}_5(h^k)), \tag{3}$$

where μ is the mean, σ^2 is the squared difference, and $N(\mu, \ln\sigma^2)$ represents the distribution of X^k. For

fitted values to contain negative numbers and to simplify the computation, VAE fits $\ln\sigma^2$, as shown in Eq. (3).

VAE employs the fitted mean and variance to represent the fitted distribution. To achieve VAE's goal, the fitted distribution constantly approximates the real distribution of X^k. The goal can be denoted as

$$\min \mathrm{KL}\left(q\left(z^k|X^k\right)\|p\left(z^k|X^k\right)\right), \tag{4}$$

where $q(z^k|X^k)$ is the fitted distribution, $p(z^k|X^k)$ is the real distribution of X^k, and KL denotes the Kullback–Leibler (K–L) divergence which measures the similarity of two probability distributions. Therefore, VAE's goal is to make $q(z^k|X^k)$ approximate $p(z^k|X^k)$.

According to the definition of K–L divergence, Eq. (4) can be transformed into

$$\begin{aligned}\min L=&\ln p\left(X^k\right)+\int q\left(z^k|X^k\right)\ln q\left(z^k|X^k\right)\mathrm{d}z^k\\&-\int q\left(z^k|X^k\right)\ln\left[\,p\left(X^k|z^k\right)p\left(z^k\right)\right]\mathrm{d}z^k,\end{aligned} \tag{5}$$

where L denotes the difference between $q(z^k|X^k)$ and $p(z^k|X^k)$, and X^k is a specific input; thus, $\ln p(X^k)$ is a fixed value. Eq. (5) can be simplified into

$$\begin{aligned}\min L=&\int q\left(z^k|X^k\right)\ln q\left(z^k|X^k\right)\mathrm{d}z^k\\&-\int q\left(z^k|X^k\right)\ln\left[\,p\left(X^k|z^k\right)p\left(z^k\right)\right]\mathrm{d}z^k.\end{aligned} \tag{6}$$

Eq. (6) can then be transformed into

$$\begin{aligned}\max\left(-L\right)=&E_{Z\sim q\left(z^k|X^k\right)}\left[\ln p\left(X^k|z^k\right)\right]\\&-\mathrm{KL}\left(q\left(z^k|X^k\right)\|p\left(z^k\right)\right),\end{aligned} \tag{7}$$

where $E_{Z\sim q\left(z^k|X^k\right)}\left[\ln p\left(X^k|z^k\right)\right]$ represents the reconstruction error and $\mathrm{KL}\left(q\left(z^k|X^k\right)\|p\left(z^k\right)\right)$ represents the regularization of VAE. Therefore, the VAE's training can be viewed as a confrontation between the encoder producing noise and the decoder resisting noise. The fitted σ^2 adds an unfixed noise that is the regularization term to boost the algorithm's resilience.

For the encoder to continuously generate noise, VAE makes the distribution fitted by the encoder satisfy the standard normal distribution. Furthermore, all $p(z|X)$'s fitted by the ideal VAE obey the standard normal distribution, $N(0, 1)$. Therefore, VAE adds an additional term to the loss function. The additional term can be denoted as

$$\mathrm{KL}\left(N\left(\mu,\sigma^2\right)\|N(0,1)\right)\right)=\frac{1}{2}\left(-\ln\sigma^2+\mu^2+\sigma^2-1\right). \tag{8}$$

However, neither the sampled μ nor σ^2 are learnable; to make the distribution denoted by μ and σ^2 learnable, VAE uses the reparameterization trick to make the distribution derivable. The calculating equation is as follows:

$$z=\varepsilon\sigma+\mu, \tag{9}$$

where ε is a random value and $\varepsilon\sim N(0, 1)$, and z is a trainable variable which is the input of the improved K-means.

From Eq. (9), we obtain the hidden variable z, and then input z into the decoder to reconstruct the real data. The calculating equation is as follows:

$$X'^k=\sigma\left(\mathrm{fc}_7\left(\sigma\left(\mathrm{fc}_6\left(z^k\right)\right)\right)\right), \tag{10}$$

where fc_6 and fc_7 are two fully connected layers that reconstruct the hidden variable $z^k\in\mathbb{R}^{N\times m}$ into $X'^k\in\mathbb{R}^{N\times n}$. The reconstructed loss of Eq. (10) can be denoted as

$$\begin{aligned}\mathrm{loss}_{\mathrm{reconstruct}}=&X\left(-\ln\left(\mathrm{sigmoid}\left(X'\right)\right)\right)\\&+\left(1-X\right)\left(-\ln\left(1-\mathrm{sigmoid}\left(X'\right)\right)\right).\end{aligned} \tag{11}$$

Combining Eqs. (8) and (11), the total loss function can be denoted as

$$\begin{aligned}\mathrm{loss}=&\frac{1}{2}\left(-\ln\sigma^2+\mu^2+\sigma^2-1\right)+X'\\&-X'X+\ln\left(1+\exp\left(-X'\right)\right).\end{aligned} \tag{12}$$

After training several batches as described above, the decoder can approximate the original data and the encoder can calculate the hidden variables we needed. At this time, the VAE's encoder can generate hidden variables with a more uniform distribution, which can handle diverse datasets smoothly and has strong robustness with low parameter needs in subsequent clustering species.

3.3.2 Server side for clustering analysis

The VAE's encoder is deployed to the vehicle's edge and, when clustering is required, the driver-side edge device computes the hidden variables and uploads

the hidden variables to the server-side. Furthermore, the server side uses the hidden variables and the improved *K*-means for clustering analysis, preventing the leakage of the user's original data and lowering server traffic. In particular, the server side uses an improved clustering algorithm. The reason for improving the *K*-means is as follows:

In *K*-means-based clustering analysis, the algorithm divides n objects into K clusters, and the final clustering results are similar within clusters and different between clusters. The cluster center is somewhat representative of the whole cluster. However, as shown in Fig. 3, the *K*-means updates the cluster center by calculating one coordinate mean of all the trajectory objects. At this time, the cluster center is an ideal state and cannot find a corresponding vehicle trajectory in the real world. As a result, with each update of the cluster center, the clustering effect becomes biased towards the ideal state and loses some meaningful value. Therefore, we make improvement on the *K*-means algorithm. The improved *K*-means algorithm is shown in Algorithm 1.

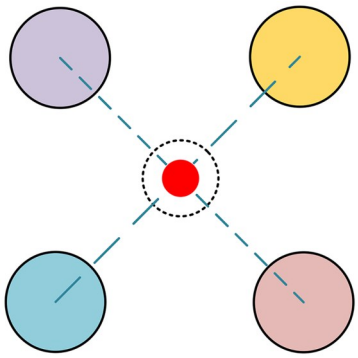

Fig. 3 *K*-means' principle of updating the cluster center

Specifically, the improved *K*-means updates the cluster center by minimizing the error value, which means that the selected cluster center is always a specific trajectory object, as shown in Fig. 4. In each cluster, after determining the included trajectory objects, the sum of the distance of each trajectory and all other trajectories can be calculated and the trajectory with the smallest sum is selected as the next cluster center. The new cluster center is calculated as follows:

$$c=\arg\min_i\left(\sum_{j=0}^{m}\mathrm{dist}\left(z_i,z_j\right)\right), \tag{13}$$

where c is the new centroid, z_i and z_j are vehicle trajectories in the same cluster, and $\mathrm{dist}(\cdot,\cdot)$ can calculate the distance between the two elements.

Algorithm 1 Improved *K*-means

Input: vehicle trajectory $X\in\mathbb{R}^{N\times d}$, the distance between vehicle trajectory $\mathrm{dist}\in\mathbb{R}^{N\times N}$

1: Initialize the number of centroids K, cluster list clusterDict=$\varnothing$, centroidIndexList (by randomly sampling K subscripts from range (0, N)), and dist (by calculating the distance between N trajectories in X)
2: **repeat**
3: **for** i=0 to N **do**
4: $j^*=\arg\min_j \mathrm{dist}[i][j]$
5: clusterDict[j^*].append(i)
6: **end for**
7: Update centroidIndexList={}
8: **for** nodeList in clusterDict **do**
9: **for** i=0 to len(nodeList) **do**
10: Update newCentroid by Eq. (13)
11: centroidIndexList.append(newCentroid)
12: **end for**
13: **end for**
14: **until** stopping criteria are met

Output: clusterDict

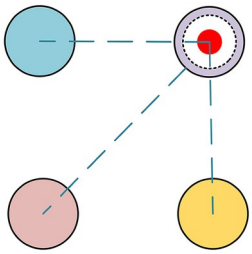

Fig. 4 Improved *K*-means' principle of updating the cluster center

In terms of run times, in the process of screening cluster centers, the larger the K is, the fewer objects there are in a cluster and the fewer the times to compare the error value of an object with those of other objects. Therefore, the run times decrease with increasing K within a certain range. The equation to calculate the run times for updating the cluster center is as follows:

$$\mathrm{Rt}=\sum_{i=1}^{K}\mathrm{size}\left(\mathrm{list}_i\right)\times\left(\mathrm{size}\left(\mathrm{list}_i\right)-1\right), \tag{14}$$

where $\mathrm{size}(\mathrm{list}_i)$ is the size of the i^{th} cluster and Rt represents the run times for updating the cluster center.

4 Experiments and analysis

4.1 Experimental settings

Dataset: We conducted a series of experiments on three real-world datasets, including the cab dataset from Shanghai, China (ShangHai), the highway vehicle dataset from Emeryville USA (Emeryville), and the lane arterial segment dataset from Los Angeles, USA (LosAngele). The ShangHai cab dataset was divided into two sub-datasets, ShangHai1 and ShangHai2, according to different time slices and different numbers of trajectories. After canvassing, sampling, and slicing the original data, any vehicle can be represented by a one-dimensional vector of length 5000.

VAE training modules: The network structure was given in detail in Section 3. The loss function used was the sigmoid-activated cross-entropy, with additional loss terms added. The total loss function was as in Eq. (12). Adopting the Adam optimizer, the learning rate was 1e−2, which was finally reduced to 8.5e−5 as the experiment progressed. The total number of epochs was 80, the batch size was 80, and the encoder is a neural network with a single hidden layer which consists of 80 neurons (in the ablation study experiment, the dimensionality of the hidden variables obtained by other dimensionality reduction algorithms was also set to 80).

Clustering module: The number of clusters K was set to $[2,29]$. For each K, the number of times the center of clusters was initialized was set to 10 (in the sensitivity analysis, it changed from 1 to 10). A total of 25 iterations were performed in each case, with the best result being taken at the end. Euclidean distance was used to measure the distance.

Metrics: Wang W et al. (2021) pointed out that artificially designed metrics may cause unexpected bias. To avoid this problem, three metrics were set in this study to evaluate the clustering results: the silhouette coefficient (SC), the Davies–Bouldin index (DBI), and the Calinski–Harabaz index (CHI). Detailed descriptions of the three metrics are as follows:

(1) SC: The formula for node v_i is as follows:

$$S(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, \tag{15}$$

where $a(i)$ is the average distance between node v_i and the other nodes in the cluster, $b(i)$ represents the average distance from node v_i to all other nodes in the nearest neighboring cluster, and the SC takes values in $[-1,1]$. The larger the SC, the better the clustering.

(2) DBI: It is calculated as follows:

$$\mathrm{DBI}=\frac{1}{N}\sum_{i=1}^{N}\max_{j\neq i}\left(\frac{\overline{S_i}+\overline{S_j}}{\|w_i-w_j\|_2}\right), \tag{16}$$

where $\overline{S_i}$ is the average distance from the object in the cluster to the cluster center, w_i is the cluster center of cluster i, $\|w_i-w_j\|_2$ is the distance between two cluster centers, and N is the number of cluster centers. DBI is the maximum value of the ratio of the sum of the average distances within all pairs of categories to the distance between two clusters. The lower the DBI, the closer the clustering results are inside clusters, the greater the gap between clusters, and the better the clustering quality.

(3) CHI: It is calculated as follows:

$$\mathrm{CHI}=\frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)}\frac{m-k}{k-1}, \tag{17}$$

where m denotes the number of nodes in the clustered object, k is the number of clusters divided, B_k is the inter-cluster covariance matrix, W_k is the intra-cluster covariance matrix, and $\mathrm{tr}(\cdot)$ is the matrix trace. The larger the CHI, the closer the clusters are within each other and the more dispersed the clusters are between each other.

4.2 Experimental results and analysis

This subsection includes four kinds of experiments: (1) comparison experiments of IKV and other clustering algorithms; (2) two ablation studies to verify the performance of two modules of IKV; (3) experiments to verify the sensitivity of the framework under different initialization numbers of cluster centers; (4) analyzing the impact of the two IKV modules on runtime.

4.2.1 Comparison study

To verify the advantage of IKV, we compared IKV with seven different baselines, including K-means (MacQueen, 1967), mini batch K-means (MBK-means) (Sculley, 2010), K-medoids (Park and Jun, 2009), spectral clustering (SPC) (Ng et al., 2001), spectral bi-clustering (SPBC) (Kluger et al., 2003), BIRCH

clustering (BIC) (Zhang et al., 1996), and FCM (Dunn, 1973; Bezdek, 1981). The number of clusters K in the experiment was set to [2, 29] and, to compare the performance of IKV with other clustering algorithms as a whole, we let K range from 2 to 29 for all the metrics and displayed the findings to two decimal places, as shown in Table 1.

As shown in Table 1, apart from the SPC algorithm, IKV has achieved comparatively good results on the majority of indicators across most datasets when compared with other comparative algorithms. Although SPC performs well on the indicator DBI, when dealing with the ShangHai1 dataset, SPC cannot complete the clustering task when $K\leq3$ due to the small number of nodes in ShangHai1. For CHI, SPC performs worse than all the other algorithms. Furthermore, IKV that obtains the best results is also based on K-means, indicating that the K-means algorithm is superior when dealing with vehicle trajectory clustering. The experimental results show that our proposed framework does not lose clustering effectiveness due to privacy-preserving goals and also improves the clustering quality to some extent. In other words, the clustering is performed with a good balance of privacy protection and utility.

Fig. 5 depicts the comparison results of IKV with other clustering methods on the corresponding number of clusters. Fig. 5 shows that the clustering effect of IKV starts to stabilize at $K>5$. The gap between IKV and other clustering algorithms becomes larger as the number of features possessed by each trajectory point in the datasets LosAngele, Emeryville, and ShangHai2 gradually decreases. As a result, IKV should be more competitive for simple datasets.

4.2.2 Ablation study

To further evaluate the effectiveness of each module in IKV, we conducted two sets of experiments to perform the verification: a comparison of VAE with other dimensionality reduction algorithms and a comparison of improved K-means with K-medoids and K-means.

To verify VAE's respective performance, we compared it with five different baselines, including PCA, AE, DictionaryLearning, SVD, and instrumented principal components analysis (IPCA). These dimensionality reduction algorithms' low-dimensional vectors were inputted into the improved K-means for clustering. We let K range from 2 to 29 for all the metrics and displayed the findings to two decimal places, as shown in Table 2.

Table 2 shows that for the dataset ShangHai2, VAE performs better than the other dimensionality reduction algorithms on SC and DBI, and has a sub-optimal performance on CHI. For the dataset LosAngele, SVD and IPCA perform well on SC and DBI, while VAE performs at an average level, but VAE performs the best on CHI.

The above illustrates the overall performance of VAE. Fig. 6 shows the exact experimental results corresponding to each K.

Fig. 6 shows that with the change of K, the performance of VAE does not change much and is relatively stable. This is because VAE solves the coding problem based on the probability distribution, and VAE generates a more uniform distribution of the hidden variables. For the dataset ShangHai2, VAE performs better on most of the K values compared to other algorithms. The combination of Table 2 and Fig. 6 shows that

Table 1 Overall comparison results of IKV and other clustering algorithms

Algorithm	SC				DBI				CHI			
	S1	S2	E	LA	S1	S2	E	LA	S1	S2	E	LA
K-means	8.95*	7.19*	1.35	2.59	21.47*	28.71	68.83	63.93	2438.09*	1684.97*	320.01	575.35
MBK-means	7.71	0.23	1.28	2.79	26.49	71.31	63.77	61.24	1895.08	997.66	366.45	616.26
K-medoids	5.04	6.42	1.28	2.54	33.21	31.39	68.10	61.10	1326.37	1429.00	349.27	584.54
SPC	–	6.64	**3.74**	2.21	–	**17.07**	**24.26**	**23.20**	–	96.67	81.00	52.52
SPBC	6.69	0.35	1.35	1.71	32.91	101.03	69.84	75.05	1379.74	1150.01	360.17	512.03
BIC	8.57	0.99	1.96*	3.37*	22.08	94.51	60.84*	56.21	2376.16	1220.71	387.62*	655.82
FCM	5.32	−0.68	0.06	**4.37**	37.98	110.12	67.92	46.95*	1308.82	746.25	345.28	**1035.02**
IKV	**10.88**	**8.00**	1.27	1.85	**20.08**	26.64*	63.64	67.03	**3734.77**	**1739.76**	**601.91**	819.19*

The best results are in bold and the second best results are marked with asterisks. SC: silhouette coefficient; DBI: Davies–Bouldin index; CHI: Calinski–Harabaz index; S1: ShangHai1; S2: ShangHai2; E: Emeryville; LA: LosAngele

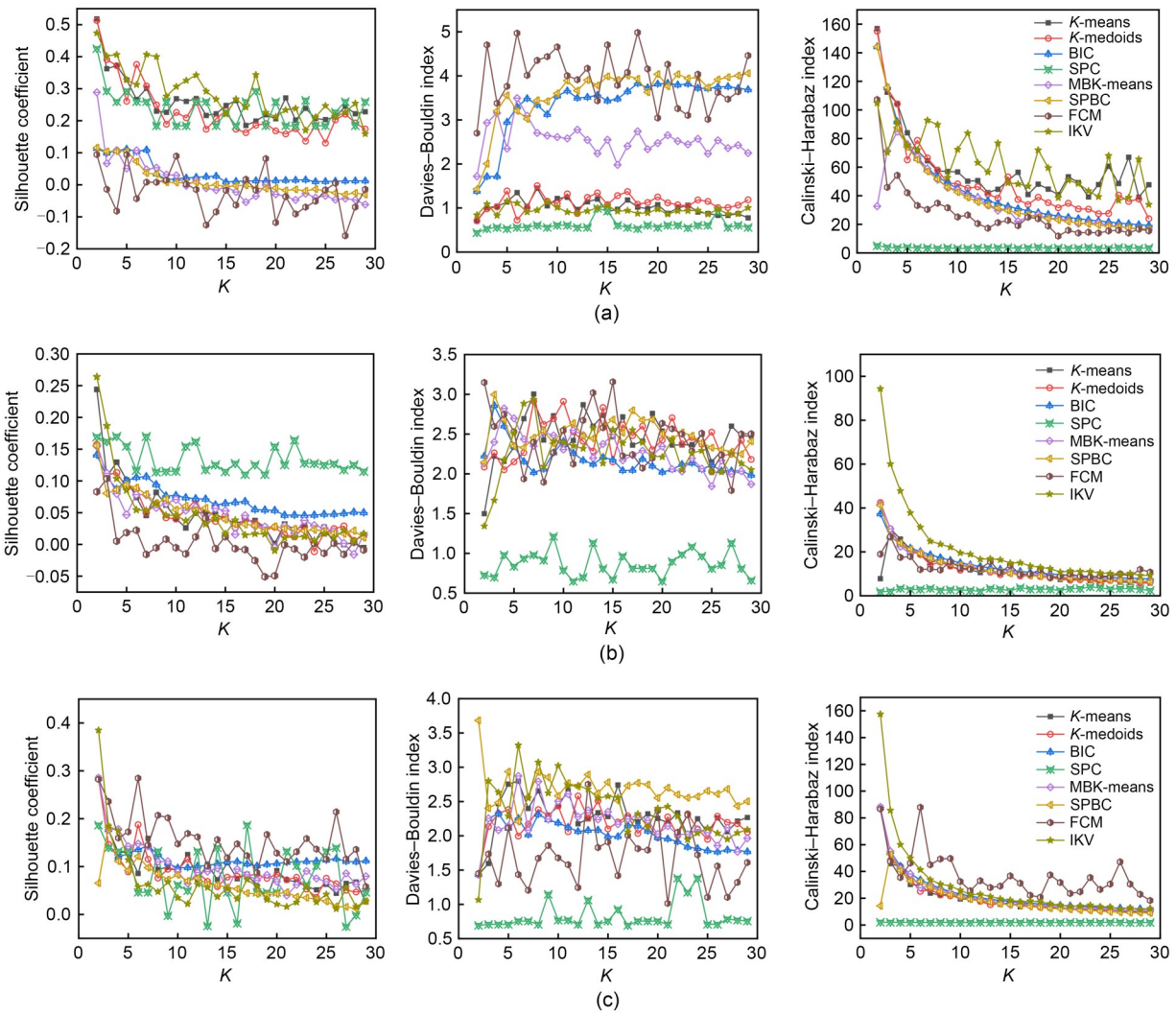

Fig. 5 Specific comparison results of IKV and other clustering algorithms: (a) ShangHai2; (b) Emeryville; (c) LosAngele

Table 2 Overall comparison results of VAE with other dimensionality reduction algorithms

Algorithm	SC		DBI		CHI	
	ShangHai2	LosAngele	ShangHai2	LosAngele	ShangHai2	LosAngele
PCA	7.43	2.62	28.28	62.29	1732.22	546.01
AE	2.77	2.18	74.66	64.55	**2534.16**	665.48
DictionaryLearning	−0.93	1.13	38.65	70.85	43.61	112.97
SVD	5.99	**2.91**	32.46	**60.60**	1372.60	644.80
IPCA	6.15	**2.91**	30.85	61.07	1341.46	647.30
VAE	**8.00**	1.85	**26.64**	67.03	1739.76	**819.19**

The best results are in bold. SC: silhouette coefficient; DBI: Davies–Bouldin index; CHI: Calinski–Harabaz index

VAE's performance is smoother and more reliable. In addition, although VAE is slightly better than other algorithms, the difference is small. This indicates that there is a wide range of algorithms available for client-side deployment. It further illustrates the flexibility of our proposed framework.

In the second ablation study, the *K*-means updates the cluster center by calculating the mean value

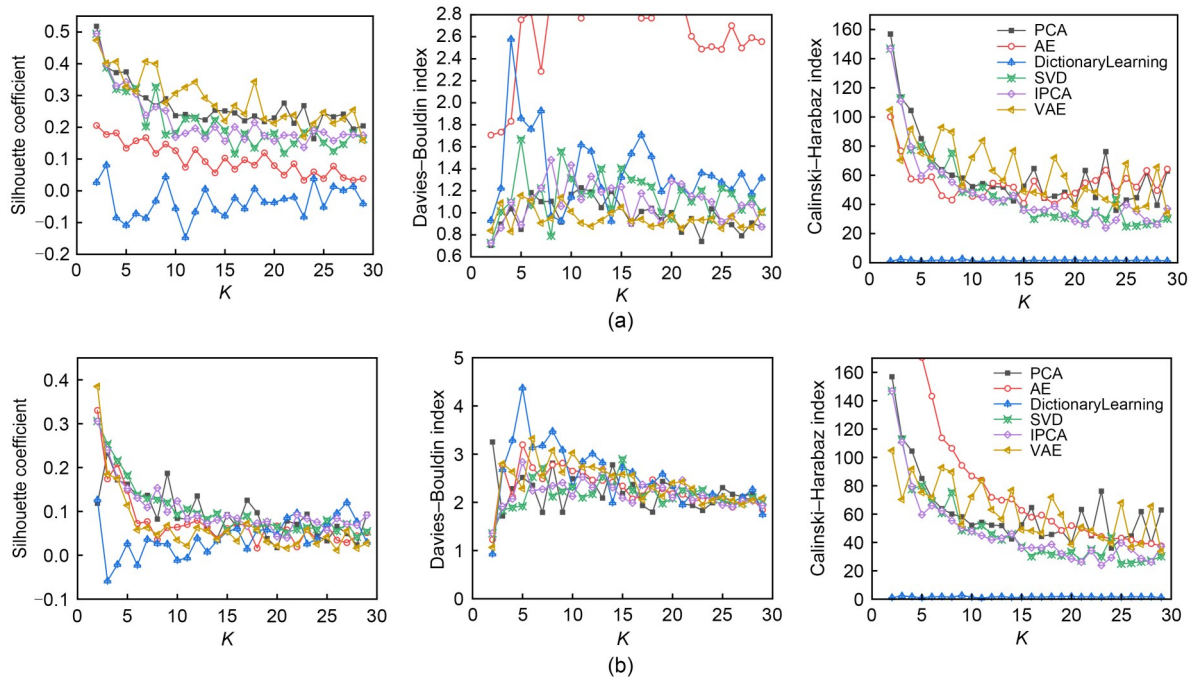

Fig. 6 Comparison results of VAE and other dimensionality reduction algorithms: (a) ShangHai2; (b) LosAngele

of each point when dealing with discrete points in coordinate space, and the clustering effect is biased toward the ideal value, but it cannot be applied to trajectory data. For handling vehicle trajectory data, the improved *K*-means has made improvements. To determine whether the clustering effect is still present on the improved basis, we compared improved *K*-means to the *K*-means algorithm and *K*-medoids based on a similar principle to *K*-means. The comparison results are shown in Fig. 7.

Fig. 7 shows that there are no appreciable differences among the clustering evaluation results of improved *K*-means, *K*-means, and *K*-medoids. When $K>15$, the clustering effect is better for most of the improved *K*-means results. This demonstrates that improved *K*-means can handle vehicle trajectory data while preserving clustering quality.

4.2.3 Sensitivity analysis

The initialization number of cluster centers can have a significant impact on the performance of *K*-means. In our research, we addressed this issue by selecting multiple sets of initial cluster centers for *K*-means and chose the best clustering result among these sets as the final output. Consequently, the initialization number can influence the clustering outcome.

To assess the effect of the initialization number of cluster centers on our proposed framework, we conducted experiments with the initial number of *K* ranging from one to nine. We performed a sensitivity analysis on the framework and evaluated its clustering performance. Fig. 8 displays the results obtained by employing different dimensionality reduction algorithms in conjunction with varying initialization number of cluster centers.

Fig. 8 shows that except for two experimental results (Figs. 8c1 and 8c2), VAE's performances are more stable in the remaining seven out of nine experimental results, highlighting the insensitivity of the data generated by VAE to variations in the initialization number of distinct cluster centers. While the CHI result of AE surpasses that of VAE on the ShangHai2 dataset, the stability of AE's performance falls short in comparison to that of VAE. This observation suggests that VAE is better suited for IKV due to its higher stability and robustness.

4.2.4 Time complexity analysis

Because the improved *K*-means algorithm can calculate the distance between each trajectory in advance, it significantly outperforms the *K*-means algorithm in terms of the algorithm runtime. Since *K*-means is too

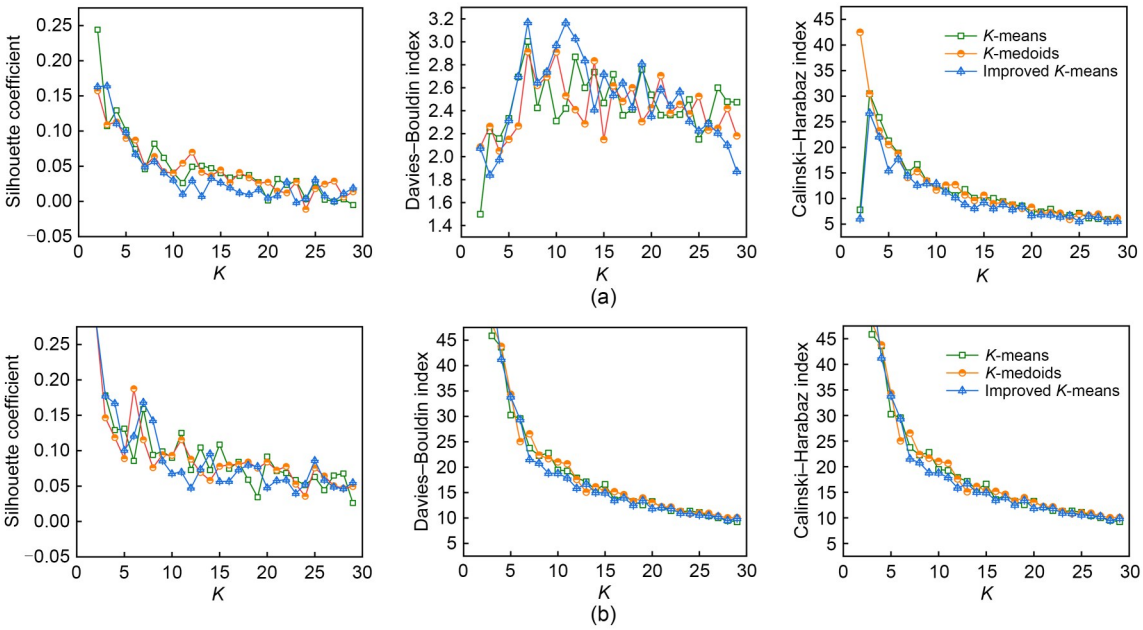

Fig. 7 Comparison results of improved *K*-means, *K*-means, and *K*-medoids: (a) Emeryville; (b) LosAngele

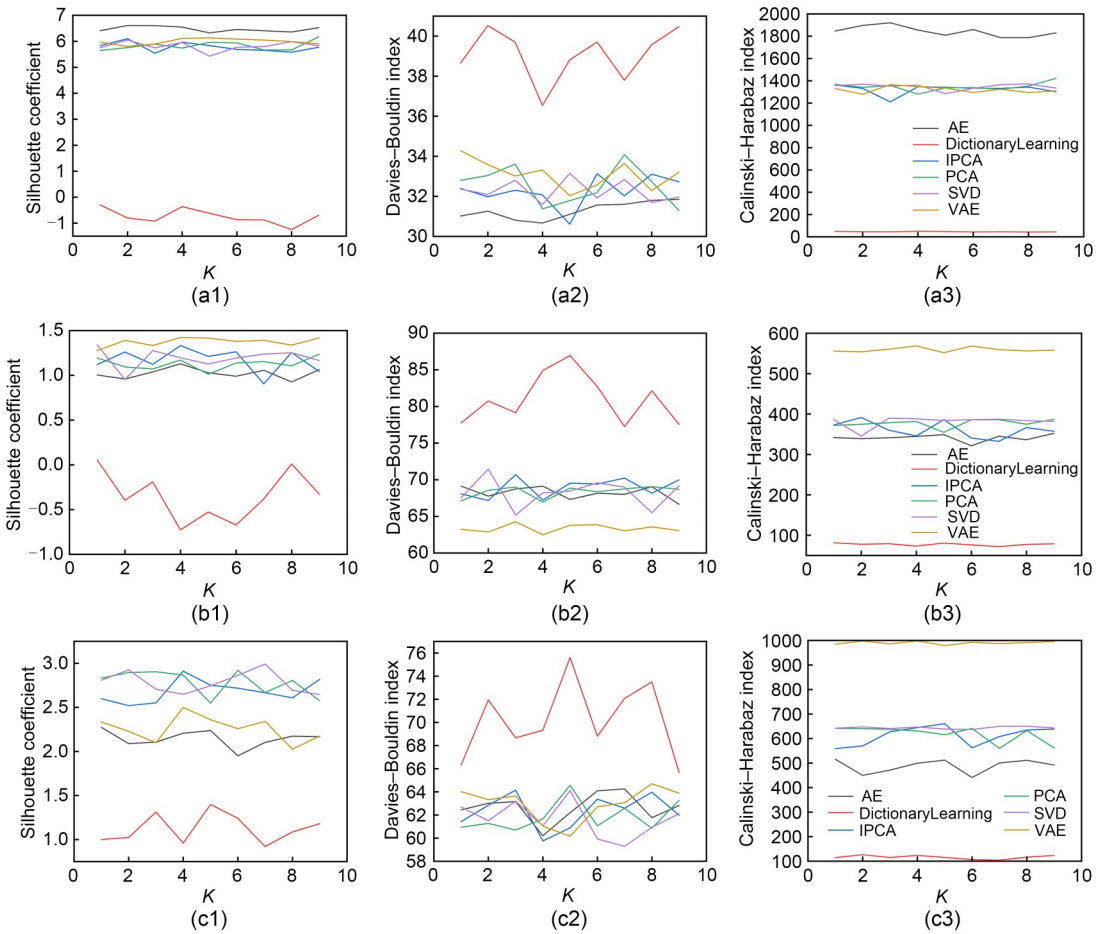

Fig. 8 Comparison results under different initialization number of cluster centers: (a1–a3) ShangHai2; (b1–b3) Emeryville; (c1–c3) LosAngele

time-consuming, to highlight the trend of the runtime of the improved *K*-means and the *K*-means as *K* increases, Fig. 9 shows the runtime comparison among IK, KV, and IKV.

Fig. 9 illustrates two points:

(1) Comparing IK and IKV, it is clear that the runtime of VAE-processed improved *K*-means is lower than that of improved *K*-means, indicating that VAE encoding of trajectory data can significantly reduce computational cost.

(2) Comparing KV and IKV, the larger the *K*, the faster the improved *K*-means running and the slower the *K*-means running. At *K*=11, even though no low-dimensional embedding vectors are used, the runtime of improved *K*-means begins to be lower than that of the *K*-means algorithm employing low-dimensional embedding vectors. Combined with Eq. (14), it is clear that the improved *K*-means algorithm's running cost will be more advantageous when *K* is larger.

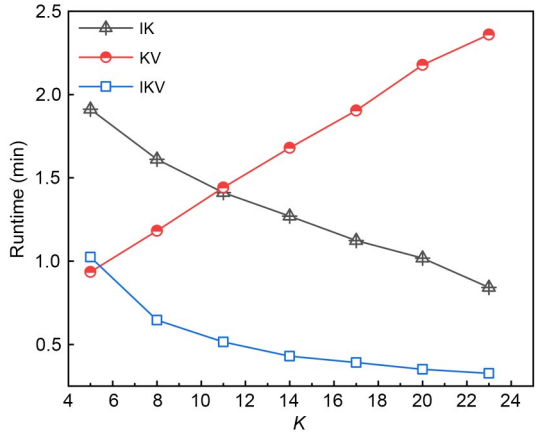

Fig. 9 Runtime comparison

5 Conclusions and future work

With the development of edge cloud computing, smart transportation has once again been pushed to new heights. However, the development of smart transportation inevitably brings challenges in terms of privacy and security. Attackers can use vehicle trajectory data to obtain sensitive information. Trajectory clustering is one of the methods for mining vehicle behavioral patterns. To prevent privacy leakage caused by uploading original user data, we proposed a privacy-preserving vehicle trajectory clustering framework. We conducted extensive experiments for models deployed at the edge of the vehicle in the framework and discovered that IKV performed well. The three metrics from the experiments showed that IKV can handle the special data of vehicle trajectory and significantly improve the clustering effect while protecting privacy. Moreover, due to the low sensitivity to the number of cluster centers, the proposed framework can be applied to clustering tasks of different magnitudes.

However, if the decoder information of the model's VAE is leaked, the original data may be reverted to trajectory data that approximate the original data. Further encryption of the decoder is a problem that must be addressed in the future to avoid the privacy concern posed by model information leakage.

Contributors

Ran TIAN: conceptualization, supervision, methodology, project administration, writing–review and editing. Pulun GAO: methodology, data curation, software, writing–original draft, visualization, investigation, validation. Yanxing LIU: supervision, project administration.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Advani C, Bhaskar A, Haque MM, 2022. Bi-level clustering of vehicle trajectories for path choice set and its nested structure identification. *Transp Res Part C Emerg Technol*, 144:103895. https://doi.org/10.1016/j.trc.2022.103895

Arif M, Chen JE, Wang GJ, et al., 2021. Privacy preserving and data publication for vehicular trajectories with differential privacy. *Measurement*, 173:108675. https://doi.org/10.1016/j.measurement.2020.108675

Atev S, Miller G, Papanikolopoulos NP, 2010. Clustering of vehicle trajectories. *IEEE Trans Intell Transp Syst*, 11(3): 647-657. https://doi.org/10.1109/tits.2010.2048101

Benkert M, Gudmundsson J, Hübner F, et al., 2008. Reporting flock patterns. *Comput Geomet*, 41(3):111-125. https://doi.org/10.1016/j.comgeo.2007.10.003

Besse PC, Guillouet B, Loubes JM, et al., 2016. Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Trans Intell Transp Syst*, 17(11):3306-3317. https://doi.org/10.1109/tits.2016.2547641

Bezdek JC, 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Springer, New York, USA. https://doi.org/10.1007/978-1-4757-0450-1

Caliński T, Harabasz J, 1974. A dendrite method for cluster analysis. *Commun Statist*, 3(1):1-27. https://doi.org/10.1080/03610927408827101

Chang LC, Chang FJ, Yang SN, et al., 2020. Self-organizing maps of typhoon tracks allow for flood forecasts up to two days in advance. *Nat Commun*, 11(1):1983. https://doi.org/10.1038/s41467-020-15734-7

Chen CM, Ye Z, Hu F, et al., 2021. Vehicle trajectory-clustering method based on road-network-sensitive features. *J Intell Fuzzy Syst*, 41(1):2357-2375. https://doi.org/10.3233/jifs-211270

Cho K, Van Merriënboer B, Gulcehre C, et al., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. https://doi.org/10.48550/arxiv.1406.1078

Davies DL, Bouldin DW, 1979. A cluster separation measure. *IEEE Trans Patt Anal Mach Intell*, PAMI-1(2):224-227. https://doi.org/10.1109/tpami.1979.4766909

Dunn JC, 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J Cybern*, 3(3):32-57. https://doi.org/10.1080/01969727308546046

Gaffney S, Smyth P, 1999. Trajectory clustering with mixtures of regression models. Proc 5th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.63-72. https://doi.org/10.1145/312129.312198

Gariel M, Srivastava AN, Feron E, 2011. Trajectory clustering and an application to airspace monitoring. *IEEE Trans Intell Transp Syst*, 12(4):1511-1524. https://doi.org/10.1109/tits.2011.2160628

Gazdag A, Lestyán S, Remeli M, et al., 2023. Privacy pitfalls of releasing in-vehicle network data. *Veh Commun*, 39: 100565. https://doi.org/10.1016/j.vehcom.2022.100565

Guo MM, Jin XY, Pissinou N, et al., 2015. In-network trajectory privacy preservation. *ACM Comput Surv*, 48(2):23. https://doi.org/10.1145/2818183

Hong ZH, Chen Y, Mahmassani HS, et al., 2017. Commuter ride-sharing using topology-based vehicle trajectory clustering: methodology, application and impact evaluation. *Transp Res Part C Emerg Technol*, 85:573-590. https://doi.org/10.1016/j.trc.2017.10.020

Hong ZH, Chen Y, Mahmassani HS, 2018. Recognizing network trip patterns using a spatio-temporal vehicle trajectory clustering algorithm. *IEEE Trans Intell Transp Syst*, 19(8): 2548-2557. https://doi.org/10.1109/tits.2017.2754401

Jin FM, Hua W, Francia M, et al., 2023. A survey and experimental study on privacy-preserving trajectory data publishing. *IEEE Trans Knowl Data Eng*, 35(6):5577-5596. https://doi.org/10.1109/tkde.2022.3174204

Kluger Y, Basri R, Chang JT, et al., 2003. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res*, 13(4):703-716. https://doi.org/10.1101/gr.648603

Kong XP, Lu L, 2022. Privacy-preserved vehicular trajectory embedding federated learning and clustering. *J Nanjing Norm Univ (Eng Technol Ed)*, 22(2):80-86 (in Chinese). https://doi.org/10.3969/j.issn.1672-1292.2022.02.012

Lu L, Lin Y, Wen Y, et al., 2023. Federated clustering for recognizing driving styles from private trajectories. *Eng Appl Artif Intell*, 118:105714. https://doi.org/10.1016/j.engappai.2022.105714

Ma Z, Zhang T, Liu XM, et al., 2019. Real-time privacy-preserving data release over vehicle trajectory. *IEEE Trans Veh Technol*, 68(8):8091-8102. https://doi.org/10.1109/tvt.2019.2924679

MacQueen J, 1967. Some methods for classification and analysis of multivariate observations. Proc 5th Berkeley Symp on Mathematical Statistics and Probability, p.281-297.

Ng AY, Jordan MI, Weiss Y, 2001. On spectral clustering: analysis and an algorithm. Proc 14th Int Conf on Neural Information Processing Systems: Natural and Synthetic, p.849-856.

Oksanen J, Bergman C, Sainio J, et al., 2015. Methods for deriving and calibrating privacy-preserving heat maps from mobile sports tracking application data. *J Transp Geogr*, 48:135-144. https://doi.org/10.1016/j.jtrangeo.2015.09.001

Pan XL, Wang HP, Cheng XQ, et al., 2020. Online detection of anomaly behaviors based on multidimensional trajectories. *Inform Fusion*, 58:40-51. https://doi.org/10.1016/j.inffus.2019.12.009

Park HS, Jun CH, 2009. A simple and fast algorithm for *K*-medoids clustering. *Expert Syst Appl*, 36(2):3336-3341. https://doi.org/10.1016/j.eswa.2008.01.039

Rousseeuw PJ, 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math*, 20:53-65. https://doi.org/10.1016/0377-0427(87)90125-7

Sculley D, 2010. Web-scale *k*-means clustering. Proc 19th Int Conf on World Wide Web, p.1177-1178. https://doi.org/10.1145/1772690.1772862

Wang H, Xu ZQ, 2018. Differential privacy preserving method for trajectory clustering. *J Huazhong Univ Sci Technol (Nat Sci Ed)*, 46(1):32-36 (in Chinese). https://doi.org/10.13245/j.hust.180107

Wang W, Xia F, Nie HS, et al., 2021. Vehicle trajectory clustering based on dynamic representation learning of Internet of Vehicles. *IEEE Trans Intell Transp Syst*, 22(6):3567-3576. https://doi.org/10.1109/tits.2020.2995856

Xie XW, Wu B, Hou BT, 2023. BEPHAP: a blockchain-based efficient privacy-preserving handover authentication protocol with key agreement for Internet of Vehicles. *J Syst Archit*, 138:102869. https://doi.org/10.1016/j.sysarc.2023.102869

Xin Y, Xie ZQ, Yang J, 2017. The privacy preserving method for dynamic trajectory releasing based on adaptive clustering. *Inform Sci*, 378:131-143. https://doi.org/10.1016/j.ins.2016.10.038

Xu XH, Liu CS, Li JH, et al., 2022. Trajectory clustering for SVR-based time of arrival estimation. *Ocean Eng*, 259: 111930. https://doi.org/10.1016/j.oceaneng.2022.111930

Yi DW, Su JY, Liu CJ, et al., 2019. Trajectory clustering aided personalized driver intention prediction for intelligent vehicles. *IEEE Trans Ind Inform*, 15(6):3693-3702. https://doi.org/10.1109/tii.2018.2890141

Yu QY, Luo YL, Chen CM, et al., 2019. Trajectory similarity clustering based on multi-feature distance measurement. *Appl Intell*, 49(6):2315-2338. https://doi.org/10.1007/s10489-018-1385-x

Zhang T, Ramakrishnan R, Livny M, 1996. BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Rec*, 25(2):103-114. https://doi.org/10.1145/235968.233324

Zhao XD, Pi DC, Chen JF, 2020. Novel trajectory privacy-preserving method based on clustering using differential privacy. *Expert Syst Appl*, 149:113241. https://doi.org/10.1016/j.eswa.2020.113241

Zheng Y, 2015. Trajectory data mining. *ACM Trans Intell Syst Technol*, 6(3):29. https://doi.org/10.1145/2743025