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 †‡1, Pulun Gao †1, Yanxing Liu †1

[1]Department of Computer Science and Engineering, Northwest Normal University, Lanzhou, CO 730070 China

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

Received May. 23, 2023; Revision accepted Jul. 21, 2023; Crosschecked

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 carries the risk of privacy leakage. Therefore, one of the current challenges is determining how to perform vehicle trajectory clustering while protecting users′ privacy. We propose a privacy-preserving vehicle trajectory clustering framework. In the framework, the client calculates the hidden variables of the vehicle trajectory and uploads the variables to the server, which uses the hidden variables for clustering analysis and delivers the analysis results to the client. The specific algorithm deployed to the framework is improved K-means based on a Variational AutoEncoder (IKV). The IKV' workflow is as follows: firstly, we train the Variational AutoEncoder (VAE) with historical vehicle trajectory data. When VAE's decoder can approximate the original data, the encoder is deployed to the edge computing device; secondly, the edge device transmits the hidden variable to the server; finally, the clustering is performed utilizing improved K-means, which prevents the leakage of the vehicle trajectory. IKV was compared to numerous clustering methods on three datasets. It was found that 75% of IKV's clustering results are optimal or suboptimal, and 77.78% of IKV's clustering results are more stable. Therefore, the proposed framework can be applied to privacy-conscious production environments, such as carpooling tasks. Moreover, due to the low sensitivity to the number of cluster centers, the proposed framework can be applied to clustering tasks of different magnitudes.

Unedited

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

1 Introduction

With the development of location acquisition and mobile computing technology, we can obtain a large amount of trajectory data from mobile objects 错误!未找到引用源。. Using specific information mining methods, we can extract relevant knowledge. For example, a cluster analysis of ship trajectory data can estimate ship arrival time 错误!未找到引用源。. Mining typhoon trajectories can provide early warning of flooding 错误!未找到引用源。. The analysis of multi-dimensional trajectory data allows for online monitoring of abnormal behavior 错误!未找到引用源。. This paper's research focused on vehicle trajectory. 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 错误!未找到引用源。, early driver intention prediction 错误!未找到引用源。, path choice set identification (Advani et al., 2022), ride matching for commuting trips 错误!未找到引用源。, and so on.

Vehicle trajectory clustering can be categorized into two types: (1) Vehicle trajectory clustering

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No.71961028), the Key Research and Develoment Program of Gansu(No.22YF7GA171) and the Scientific Research Project of the Lanzhou Science and Technology Program (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;
© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2020

without privacy protection. Research in this category focuses on how to improve the clustering quality of the respective studied roadway or application scenarios (Chen et al., 2021; 错误!未找到引用源。错误!未找到引用源。. However, personal information, such as user hobbies, eating habits, commuting patterns, and so on, 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; (2) Vehicle trajectory clustering with privacy protection. Research in this category focuses on how to achieve vehicle trajectory clustering while protecting user privacy. The representative studies are K-anonymity 错误!未找到引用源。, differential privacy 错误!未找到引用源。, and federated learning 错误!未找到引用源。. However, these privacy-preserving techniques may lessen clustering effects if protecting privacy.

Therefore, one of this study's challenges was to achieve privacy protection while still ensuring the usefulness of the published trajectories 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 the one hand, the original private data on vehicle trajectory are safeguarded. On the other hand, the hidden variable can significantly increase the clustering effect.

(2) The improved K-means based on Variational AutoEncoder (IKV) model is proposed for clustering vehicle trajectory based on our framework; the clustering result is improved by the hidden variables generated by the Variational AutoEncoder (VAE).

(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 IKV can be found in the real world.

(4) Experiments on three real-world datasets reveal that IKV performs well in terms of clustering effects while still protecting user privacy. Furthermore, the number of cluster center initializations has less of an impact on the framework's performance.

In this subsection, we present the background and problems of the vehicle trajectory clustering algorithm. The second subsection will provide a brief overview of existing work on vehicle trajectory clustering. The third subsection will introduce our proposed framework. The fourth subsection will present the experimental results and analysis. The fifth subsection will conclude the paper.

2 Related work

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 a time series that has 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 intervals 错误!未找到引用源。错误!未找到引用源。. The above works provided the foundation for sampling vehicle trajectory data by time domain and time intervals. Using the trajectory continuity feature, 错误!未找到引用源。 re-sampled the trajectories to generate new datasets, then subjected these time series to principal component analysis (PCA) before using them as embedding vectors in the clustering algorithm. The PCA improves clustering efficiency, but we conducted extensive experiments on various dimensionality reduction algorithms and discovered that VAE is more suitable for the clustering of road networks than PCA. Furthermore, we adjusted the strategy's structure to deploy the dimensionality reduction component to the client, preventing the server from obtaining the us-

Unedited

er's data.

Gariel's strategy works well on airplane routes but is inapplicable to complicated and varied road networks. 错误!未找到引用源。 considered road connection, road direction, 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's strategy addresses the influence of sensitive parameters on the clustering effect in the specific segment of the study. 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 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, 错误!未找到引用源。 proposed a multi-feature-based method for estimating distances and clustering vehicle trajectories based on driving speed, direction, form, continuity, and other factors. Therefore, Yu'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, 错误!未找到引用源。 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, 错误!未找到引用源。 used the silhouette coefficient 错误!未找到引用源。, the Davies Bouldin index 错误!未找到引用源。, and the Calinski–Harabaz index 错误!未找到引用源。 as metrics for evaluating clustering results, which reduced the cost of evaluating them. The algorithms above provided solutions for applying clustering algorithms in different scenarios. However, sensitive information, such as user hobbies, eating habits, commuting patterns, and so on, can be mined in vehicle tracking data. if the clustering is done directly utilizing 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 错误!未找到引用源。; 错误!未找到引用源。. Each trajectory is clustered with at least k-1 other trajectories to form an anonymous set, which is subsequently published. The attacker has only a tiny probability of capturing the correct trajectory, thus achieving privacy protection. The k-anonymous approach is static and only samples specific time segments, resulting in a significant loss of information from the vehicle trajectory data. 错误!未找到引用源。 proposed an adaptive dynamic trajectory publication algorithm that addresses the issue of inconsistency in sampling time and movement speed. Although the original user’s data are posted anonymously, they are still directly shared, so privacy protection is restricted. Further, Gazdag et al. (2023) showed that anonymization techniques can only provide erratic, empirical privacy protection for a whole population, but not worst-case privacy protection for each individual. Therefore, Gazdag promotes principled differential privacy to anonymize data. Based on differential privacy, researchers have made different contributions by combining their concerns. 错误!未找到引用源。 proposed a general differential privacy-preserving method that applies to most trajectory cluster 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, which protects the privacy of sensitive vehicle trajectories without destroying the spatio-temporal integrity of vehicle trajectories. 错误!未找到引用源。 considered the uncertainty of the dimensional correlation amongst trajectory data and limited the size of added noise, resulting in a method that successfully reduces noise and has good clustering performance. 错误!未找到引用源。 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 is leaked.

Both anonymization approaches and differen-

Unedited

tial 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 错误!未找到引用源。错误!未找到引用源。. During federation 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 upload 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 both 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 错误!未找到引用源。, 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 feature avoids the direct sharing of users' sensitive data. Furthermore, because the encoder learns the main properties of the data, it can increase clustering quality. For this approach, we conducted extensive tests using various algorithms and finally concluded that the quality of the hidden variables generated by VAE is relatively high.

3 Privacy-preserving vehicle trajectory clustering framework

3.1 The system framework

To perform vehicle trajectory clustering while protecting users' privacy and maximizing cluster effects, we propose a framework as shown in Figure 1.

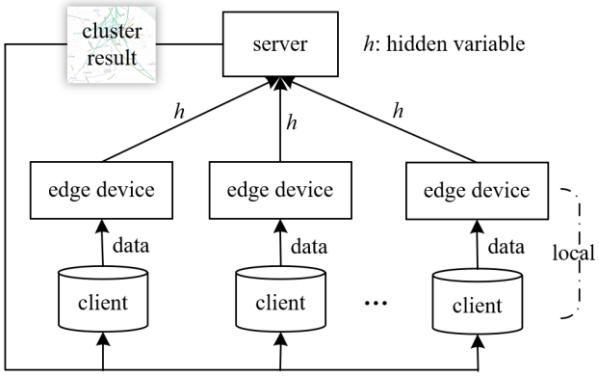


Figure 1. Privacy-preserving vehicle trajectory clustering framework.

错误!未找到引用源。 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's models, deployed by the client and server, are replaceable. (1) In the client, 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 principal component analysis (PCA), autoencoder (AE), singular value decomposition (SVD) and so on; (2) In the server, in order to classify vehicle trajectories into different clusters, the model deployed on the server can choose the clustering algorithm, such as K-means, K-mediods, fuzzy c-means (FCM), and so on.

For the two parts of the framework, the problem definition and the goals of the framework are presented in the next section.

3.2 Problem definition

3.2.1 The model deployed on the client

Unedited

Definition 1 (Vehicle trajectory data) A set $\boldsymbol{V}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_n\}$ of vehicle trajectories with a total n, where a vehicle's trajectory is defined as the trajectory point on a dynamically changeable time slice. Therefore, for any vehicle object $\boldsymbol{v}_i$, its sequence can be represented as $\boldsymbol{v}_i=(\boldsymbol{p}_{i1},\boldsymbol{p}_{i2},...,\boldsymbol{p}_{il})$, 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 $\boldsymbol{p}=(t,lon,lat,attr)$, 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_d and a decoder AE_e. The 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:

$$\min\ L(\boldsymbol{x},AE_d(AE_e(\boldsymbol{x}))) \tag{1}$$

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

Definition 3 (Hidden variables) According to Definition 1, we assume a dataset with a set $\boldsymbol{V}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_n\}$ of vehicles and $\boldsymbol{V}\in\square^{n}$. The original set of trajectories is sliced to form a unified dimension, which is then input to AE_e in Definition 2 to obtain the hidden variable $\boldsymbol{z}\in\square^{n\times d}$, which maps the original data to the latent space, and *d* is the dimension of the latent space. When 错误!未找到引用源。 satisfies certain expectations, the trajectory of ***z*** reverted through AE_e is quite similar to that of ***V***. Any trajectory object $\boldsymbol{v}_i$ in the original data can be represented by the corresponding hidden variable $\boldsymbol{z}_i$.

Based on the above definitions, the first 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 edge devices, we experimented with a large number of models. When compared to other baselines, using Variational AutoEncoder (VAE) to produce hidden variables performs better. Therefore, the client of this framework will deploy VAE. The process of producing hidden variables via VAE is explained in the next sections.

3.2.2 The model deployed on the server

Definition 4 (Clusters) We define a set $\boldsymbol{V}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_n\}$ of vehicles with a total n and a set $\boldsymbol{T}=\{\boldsymbol{t}_1,\boldsymbol{t}_2,...,\boldsymbol{t}_{n'}\}$ with a total *n'*. When $\boldsymbol{T}\in\boldsymbol{V}$, ***T*** is a cluster for ***V***.

To find vehicles with similar behavioral patterns, the server needs to divide vehicles set ***V*** into multiple clusters. The clusters set can be denoted as $C=\{\boldsymbol{T}_1,\boldsymbol{T}_2,...,\boldsymbol{T}_k\}$, where *k* is the number of clusters, and $\forall\boldsymbol{T}_i\in C,\forall\boldsymbol{T}_j\in C$, $\boldsymbol{T}_i\cap\boldsymbol{T}_j=\varnothing(i\neq j)$. Therefore, the second 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. We have improved the original K-means method, which is introduced in the next sections.

3.3 The system model

To achieve the objectives mentioned in 3.2, we assembled the better-performing Variational AutoEncoder (VAE) and improved K-means (IK) to the frame to obtain the improved K-means based on VAE (IVK). IKV's overall process is shown in 错误!未找到引用源。.

Unedited

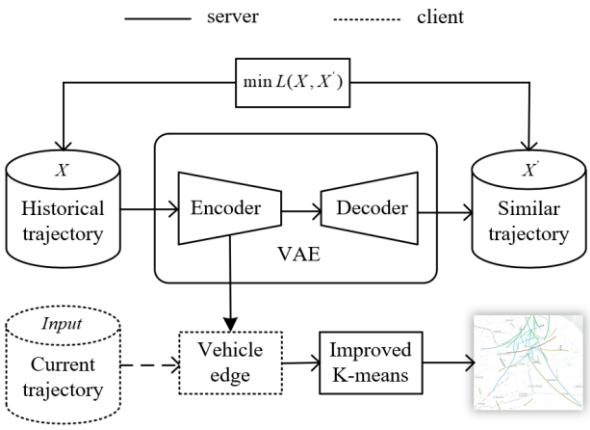


Figure 2. Overall process of IKV.

错误!未找到引用源。shows that IKV is divided into three major phases. Firstly, in the server, 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. Secondly, after the edge computing device compressed the original vehicle trajectory into the latent space hidden variable, the local edge computing device transmitted the hidden variable 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:

3.3.1 Training VAE using historical trajectories

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

$$\boldsymbol{h}^k = \sigma(fc_3(\sigma(fc_2(\sigma(fc_1(\boldsymbol{X}^k))))))) \tag{2}$$

where $\boldsymbol{X}^k \in \square^{N\times n}$ is the input data, fc_1, fc_2 and fc_3 are three fully connected layers, σ is the ReLu activation function, $\boldsymbol{h}^k$ 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 get the distribution of $\boldsymbol{X}^k$, $\boldsymbol{h}^k$ is sent in parallel to the two fully connected layers fc_4 and fc_5, which fit the mean and the squared difference of $\boldsymbol{X}^k$ simultaneously. The calculating equation is as follows:

$$N(\mu, \log\sigma^2) = N(fc_4(\boldsymbol{h}^k), fc_5(\boldsymbol{h}^k)) \tag{3}$$

where μ is the mean, σ^2 is the squared difference, $N(\mu, \log\sigma^2)$ represents the distribution of $\boldsymbol{X}^k$, and fc_4 and fc_5 are two fully connected layers. For fitted values to contain negative numbers and to simplify the computation, VAE fits the $\log\sigma^2$, 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:

$$\min KL(q(\boldsymbol{z}^k \mid \boldsymbol{X}^k) \,\|\, p(\boldsymbol{z}^k \mid \boldsymbol{X}^k)) \tag{4}$$

where $q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)$ is the fitted distribution, $p(\boldsymbol{z}^k \mid \boldsymbol{X}^k)$ is the real distribution of $\boldsymbol{X}^k$, and KL denotes the K-L scatter which measures the similarity of two probability distributions. Therefore, VAE's goal is to make $q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)$ approximate $p(\boldsymbol{z}^k \mid \boldsymbol{X}^k)$.

According to the definition of K-L scatter, Eq. (4) can be transformed into:

$$\begin{aligned}\min L = \log p(\boldsymbol{X}^k) + \int q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)\log q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)dz^k \\ -\int q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)\log[p(\boldsymbol{X}^k \mid \boldsymbol{z}^k)p(\boldsymbol{z}^k)]d\boldsymbol{z}^k\end{aligned} \tag{5}$$

where *L* denotes the difference between $q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)$ and $p(\boldsymbol{z}^k \mid \boldsymbol{X}^k)$, and $\boldsymbol{X}^k$ is a specific input, thus $\log p(\boldsymbol{X}^k)$ is a fixed value. Eq. (5) can be simplified into:

$$\begin{aligned}\min L = \int q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)\log q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)dz^k \\ -\int q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)\log[p(\boldsymbol{X}^k \mid \boldsymbol{z}^k)p(\boldsymbol{z}^k)]d\boldsymbol{z}^k\end{aligned} \tag{6}$$

Eq. (6) can then be transformed into:

$$\begin{aligned}\max(-L) = E_{Z\sim q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k \mid \boldsymbol{z}^k)] \\ -KL(q(\boldsymbol{z}^k \mid \boldsymbol{X}^k) \,\|\, p(\boldsymbol{z}^k))\end{aligned} \tag{7}$$

where $E_{Z\sim q(\boldsymbol{z}^k \mid \boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k \mid \boldsymbol{z}^k)]$ represents the reconstruction error and $KL(q(\boldsymbol{z}^k \mid \boldsymbol{X}^k) \,\|\, p(\boldsymbol{z}^k))$

Unedited

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. Further, all $p(\boldsymbol{z}\,|\,\boldsymbol{X})$ fitted by the ideal VAE obey the standard normal distribution, $\mathrm{N}(0,1)$. Therefore, VAE adds an additional term to the loss function. The additional term can be denoted as:

$$KL(N(\mu,\sigma^2 \square N(0,1))) = \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1) \quad (8)$$

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:

$$z = \varepsilon \times \sigma + \mu \quad (9)$$

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

From Eq. (9), we get the hidden variable $\boldsymbol{z}$, then input $\boldsymbol{z}$ into the decoder to reconstruct the real data. The calculating equation is as follows:

$$\boldsymbol{X}'^{\boldsymbol{k}} = \sigma(fc_7(\sigma(fc_6(\boldsymbol{z}^{\boldsymbol{k}})))) \quad (10)$$

where fc_6 and fc_7 are two fully connected layers that reconstruct the hidden variable $\boldsymbol{z}^{\boldsymbol{k}} \in \square^{N*m}$ into $\boldsymbol{X}'^{\boldsymbol{k}} \in \square^{N\times n}$. The $\boldsymbol{X}'^{\boldsymbol{k}} = \sigma(fc_7(\sigma(fc_6(\boldsymbol{z}^{\boldsymbol{k}}))))$ reconstructed loss can be denoted as:

$$loss_{resconstruct} = \boldsymbol{X} * (-)\log(sigmoid(\boldsymbol{X}')) + (1-\boldsymbol{X}) * (-)\log(1 - sigmoid(\boldsymbol{X}')) \quad (11)$$

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

$$loss = \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1) + \boldsymbol{X}' - \boldsymbol{X}' * \boldsymbol{X} + \log(1 + \exp(-\boldsymbol{X}')) \quad (12)$$

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

3.3.2 Server-side for cluster analysis

The VAE encoder is deployed to the vehicle's edge and, when clustering is required, the driver-side edge device computes the hidden variable and uploads the hidden variable to the server-side. Further, the server-side uses the hidden variable 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 cluster 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 Figure 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 improved on the K-means algorithm. The improved K-means algorithm is shown in Algorithm 1.

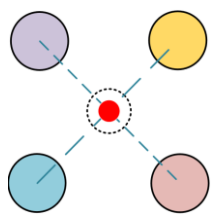


Figure 3. K-means' principle of updating the cluster center

Specifically, the improved K-means updates the cluster center by minimizing the error value, which means the selected cluster center is always a specific trajectory object, as shown in Figure 4. In each cluster, after determining the included trajectory

Unedited

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 Eq. (13).

$$c = \arg\min_i \left(\sum_{j=0}^{m} dist(\boldsymbol{z}_i, \boldsymbol{z}_j)\right) \tag{13}$$

where c is the new centroid, $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are vehicle trajectories in the same cluster, and $dist(\cdot|\cdot)$ can calculate the distance between the two scalars.

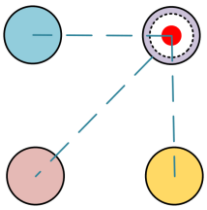


Figure 4. Improved K-means' principle of updating the cluster center.

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

$$Rt = \sum_{i=1}^{k} size(list_i) \times (size(list_i) - 1) \tag{14}$$

where $size(list_i)$ is the size of the *i*-th cluster and *Rt* is the running time for updating the cluster center.

Algorithm 1 improved K-means

Input: Vehicle Trajectory $\boldsymbol{X} \in \square^{N \times D}$, the distance between vehicle Trajectory $\boldsymbol{dist} \in \square^{N \times N}$

Output: $clusterDict$

1: initialize the number of centroids *K*, cluster list $clusterDict = \varnothing$, *scentroidIndexList* = Randomly sample *K* subscripts from range(0, N).
2: **repeat**
3: **for** *i* = 0 to N **do**
4: update by allocating point *i* to the cluster with the nearest cluster center in $scentroidIndexList$
5: **end for**
6: update $clusterDict = \varnothing$, $scentroidIndexList$ = {}
7: **for** $ndoeList$ in $clusterDict$ **do**
8: **for** *i* = 0 to len(*nodeList*) **do**
9: update *newCentroid* by Eq. (13)
10: *scentroidIndexList*.append(*newCentroid*)
11: **end for**
12: **end for**
13: **until** stopping criteria is met.

4 Experiment 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 (Emeryville), USA, and the Lane arterial segment dataset from Los Angeles, USA (LosAngele). The Shanghai Cab dataset is 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's training modules: The network structure is 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-5, which was finally reduced to 8.5e-5 as the experiment progressed. The total epoch was 80 and stopped at the appropriate time, the batch size was 80, and the final network size of the encoder was 80, which controls the size of the data uploaded to the server (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 full 25 iterations were performed in each case, with the best result being taken at the end. Euclidean distance was used to measure distance.

Metrics: Wang [20] pointed out that artificially designed metrics may cause unexpected bias; to

Unedited

avoid this problem, three metrics were set in this paper to evaluate the clustering results: the Silhouette Coefficient, the Davies–Bouldin index, and the Calinski–Harabaz index. Detailed descriptions of the three metrics are as follows:

(1) Silhouette Coefficient: The formula for the Silhouette Coefficient for a 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 a node v_i and the other nodes in the cluster, $b(i)$ is the shortest average distance between node v_i and all other nodes, and the Silhouette Coefficient takes values between [-1,1]. The larger the Silhouette Coefficient, the better the clustering.

(2) Davies–Bouldin index: The Davies–Bouldin index is calculated as follows:

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

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 the cluster i, $\|w_i-w_j\|_2$ is the distance between two cluster centers, and N is the number of cluster centers. The Davies–Bouldin index 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 Davies–Bouldin index value, the closer the clustering results are inside clusters, the greater the gap between clusters, and the better the clustering quality.

(3) Calinski–Harabaz index: The Calinski-Harabaz index is calculated as follows:

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

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 tr is the matrix trace. The larger the Calinski–Harabaz index value, 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 section includes four kinds of experiments: (1) Comparison experiment of IKV and other clustering algorithms; (2) Two ablation experiments to verify the performance of two modules of IKV; (3) Experiment to verify the sensitivity of the framework under different initialization numbers of cluster centers; (4) Analyzing the impact of the two IKV modules on the running time.

4.2.1 Comparison study

To verify the advantage of IKV, we compared IKV with eight different baselines, including: K-means (MacQueen, 1967), mini batch K-means (MBK-means) (Sculley, 2010), K-medoids (Park and Jun, 2009), spectral clustering (SPC) (Andrew Y et al., 2001), spectral bi-clustering (SPBC) (Kluger et al., 2003), birch clustering (BIC) (Zhang et al., 1996), and fuzzy c-means (FCM) (Dunn, 1973; J.C., 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 totaled all the metrics for *K* ranging from 2 to 29 and displayed the findings to two decimal places, as shown in

Table 1. Overall comparison experiment of IKV and other clustering algorithms.

Dataset	ShangHai1			ShangHai2			Emeryville			LosAngele		
Metrics	SC	DBI	CHI	SC	DBI	CHI	SC	DBI	CHI	SC	DBI	CHI
K-means	8.95*	21.47*	2,438.09*	7.19*	28.71	1,684.97*	1.35	68.83	320.01	2.59	63.93	575.35
MBK-means	7.71	26.49	1,895.08	0.23	71.31	997.66	1.28	63.77	366.45	2.79	61.24	616.26
K-medoids	5.04	33.21	1,326.37	6.42	31.39	1,429.00	1.28	68.10	349.27	2.54	61.10	584.54
SPC	-	-	-	6.64	**17.07**	96.67	**3.74**	**24.26**	81.00	2.21	**23.20**	52.52
SPBC	6.69	32.91	1,379.74	0.35	101.03	1,150.01	1.35	69.84	360.17	1.71	75.05	512.03
BIC	8.57	22.08	2,376.16	0.99	94.51	1,220.71	1.96*	60.84*	387.62*	3.37*	56.21	655.82
FCM	5.32	37.98	1,308.82	-0.68	110.12	746.25	0.06	67.92	345.28	**4.37**	46.95*	**1,035.02**
IKV	**10.88**	**20.08**	**3,734.77**	**8.00**	26.64*	**1,739.76**	1.27	63.64	**601.91**	1.85	67.03	819.19*

* The best result is in bold, and the second best result is marked with an asterisk

Table 1. Overall comparison experiment of IKV and other clustering algorithms.

Dataset	ShangHai1			ShangHai2			Emeryville			LosAngele		
Metrics	SC	DBI	CHI	SC	DBI	CHI	SC	DBI	CHI	SC	DBI	CHI

Unedited

K-means	8.95*	21.47*	2,438.09*	7.19*	28.71	1,684.97*	1.35	68.83	320.01	2.59	63.93	575.35
MBK-means	7.71	26.49	1,895.08	0.23	71.31	997.66	1.28	63.77	366.45	2.79	61.24	616.26
L-medoids	5.04	33.21	1,326.37	6.42	31.39	1,429.00	1.28	68.10	349.27	2.54	61.10	584.54
SPC	-	-	-	6.64	**17.07**	96.67	**3.74**	**24.26**	81.00	2.21	**23.20**	52.52
SPBC	6.69	32.91	1,379.74	0.35	101.03	1,150.01	1.35	69.84	360.17	1.71	75.05	512.03
BIC	8.57	22.08	2,376.16	0.99	94.51	1,220.71	1.96*	60.84*	387.62*	3.37*	56.21	655.82
FCM	5.32	37.98	1,308.82	-0.68	110.12	746.25	0.06	67.92	345.28	**4.37**	46.95*	**1,035.02**
IKV	**10.88**	**20.08**	**3,734.77**	**8.00**	26.64*	**1,739.76**	1.27	63.64	**601.91**	1.85	67.03	819.19*

* The best result is in bold, and the second best result is marked with an asterisk

As shown in Table 1, except for the Spectral algorithm, which outperforms KV on the metric DBI, and a few SC metrics where IKV performs slightly worse, we propose that KV substantially outperforms the other algorithms on all datasets. Although the Spectral algorithm performs well on the indicator CHI, when dealing with the ShangHai1 dataset, Spectral cannot complete the clustering task when K <= 3 due to the small number of nodes in ShangHai1. For CHI, the Spectral algorithm performs worse than all the other algorithms. Furthermore, K-means achieves the majority of the poor outcomes at various baselines, and the 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.

Looking at Table 1 as a whole, the clustering effect of IKV becomes worse as the number of features in the dataset increases (the table moves from left to right and the features recorded in the dataset gradually increase). This demonstrates that IKV is better suited for datasets with fewer features. The above illustrates the overall performance of IKV. Figure 5 depicts the experimental results of IKV with other clustering methods on the corresponding number of clusters.

Figure 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 features possessed by each trajectory point in the datasets LosAngele, Emeryville, and ShangHai2 gradually decrease. As a result, IKV should be more competitive for simple datasets.

4.2.2 Ablation study

To further evaluate the effectiveness of each modules 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.

In order to verify VAE's respective performance, we compared VAE with eight different baselines, including: Principal Component Analysis (PCA), Auto Encoder (AE), DictionaryLearning, Singular Value Decomposition (SVD), and Instrumented Principal Components Analysis (IPCA). These dimensionality reduction algorithms' low-dimensional vectors were input into the improved K-means for clustering. We totaled all the metrics for *K* from 2 to 29 and displayed the findings to two decimal places, as shown in Table 2.

Unedited

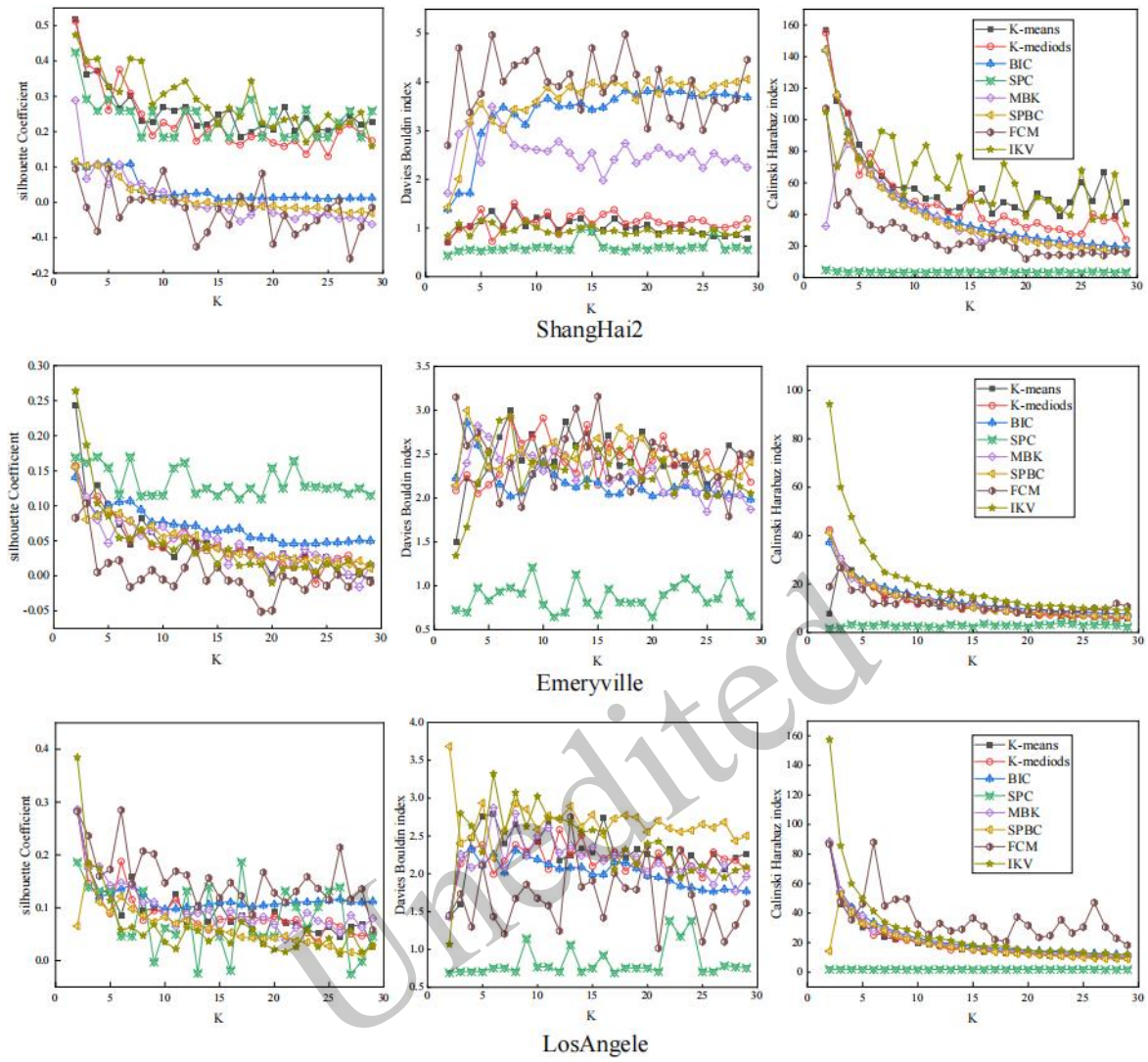


Figure 5. Specific comparison experiment of IKV and other clustering algorithms.

Unedited

Table 2. Overall comparison experiment of VAE with other dimensionality reduction algorithms.

Dataset	ShangHai2			LosAngele		
Metrics	SC	DBI	CHI	SC	DBI	CHI
PCA	7.43	28.28	1732.22	2.62	62.29	546.01
AE	2.77	74.66	**2,534.16**	2.18	64.55	665.48
DictionaryLearning	-0.93	38.65	43.61	1.13	70.85	1,12.97
SVD	5.99	32.46	1,372.60	**2.91**	**60.60**	644.80
IPCA	6.15	30.85	1,341.46	**2.91**	61.07	647.30
VAE	**8.00**	**26.64**	1,739.76	1.85	67.03	**819.19**

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, it performs best on SC and CHI. For the dataset LosAngele, SVD and IPCA performed well on SC and DBI, while VAE performed at an average level, but VAE performed best on CHI.

The above illustrates the overall performance of VAE. Figure 6 shows the exact experimental re-

sults corresponding to each *K*.

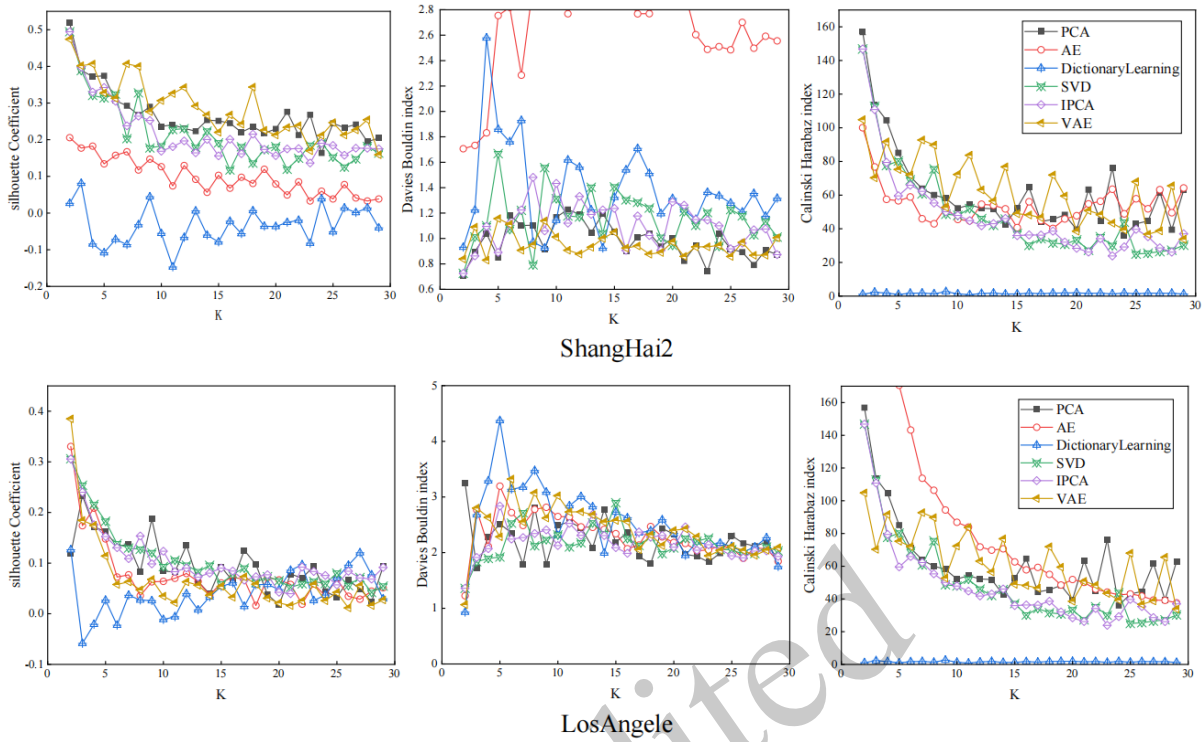


Figure 6. Comparison experiment of VAE and other dimensionality reduction algorithms.

Figure 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* compared to other algorithms. The combination of Table 2 and Figure 6 shows that VAE's performance is smoother and more reliable. In addition, although VAE is relatively 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 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 experimental results are shown in Figure 7.

Figure 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 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 number of initializations can influence the clustering outcome.

To assess the effect of the number of cluster initializations on our proposed framework, we conducted experiments with the initial number of *K*

Unedited

ranging from one to nine. We performed a sensitivity analysis on the framework and evaluated its clustering performance. Figure 8 displays the results obtained by employing different dimensionality reduction algorithms in conjunction with varying numbers of initializations.

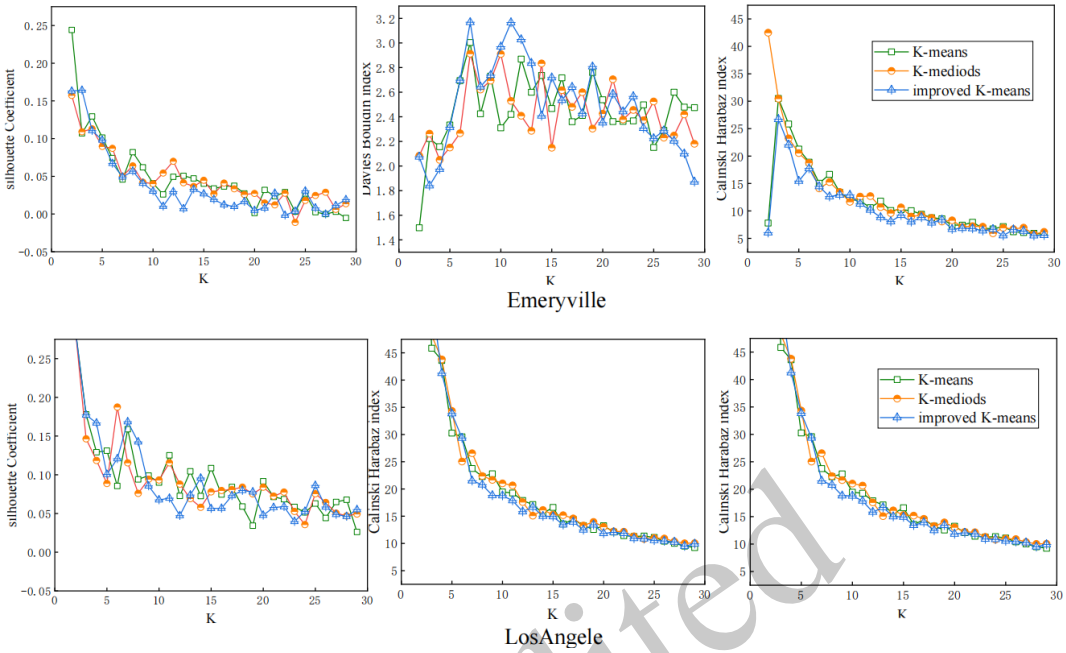
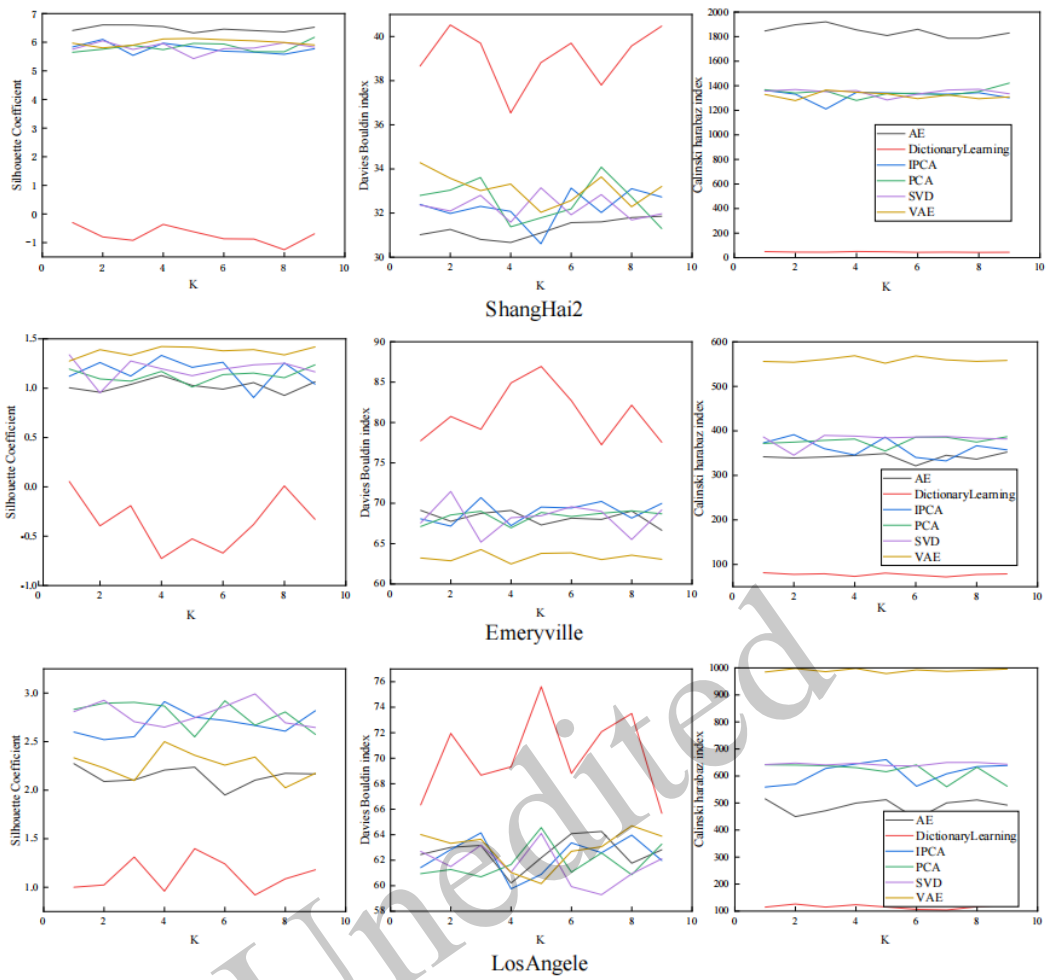


Figure 7. Comparison experiment of improved K-means with K-means and K-medoids.

Unedited



Unedited

Figure 8. Comparison experiment with different initialization numbers.

Figure 8 shows that, except for the DBI metrics on Shanghai2 and LosAngele, VAE's performances are more stable in the remaining seven-ninth of experimental results, highlighting the insensitivity of the data generated by VAE to variations in the number of initializations 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 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 running time. Since K-means is too time-consuming, to highlight the trend of the running time of the improved K-means and the K-means as *K* increases, Figure 9 does not show the K-means running time but instead presents the time complexity of K-means and improved K-means with IKV and KV.

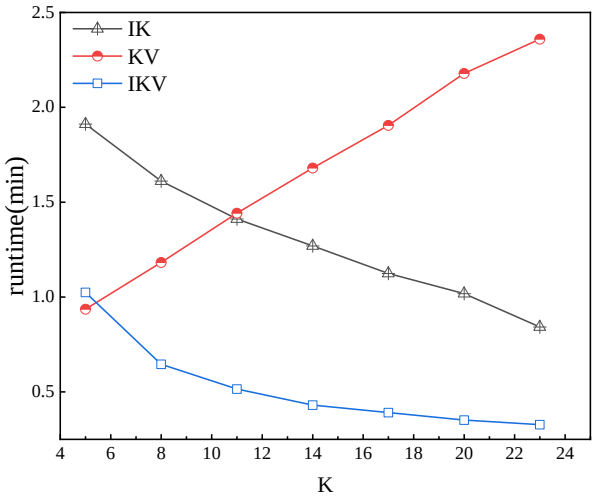


Figure 9. Runtime comparison.

Figure 9 illustrates two points:

(1) Comparing IK and IKV, it is clear that the running time of VAE-processed improved K-means is less 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 run and the slower the K-means run. At K = 11, even though no low-dimensional embedding vectors are utilized, the running time of improved K-means begins to be less 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.

5 Conclusion 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 users' 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 the improved K-means based on the Variational AutoEncoder (IKV) performed well. The three metrics from the experiments show 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 & editing. Pulun Gao: Methodology, Data curation, Software, Writing – original draft, Visualization, Investigation, Validation. Yanxing Liu: Supervision, Project administration.

Compliance with ethics guidelines

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Zheng, Y., 2015. Trajectory data mining. *ACM Transactions on Intelligent Systems and Technology*, *6*(3):1–41. https://doi.org/10.1145/2743025

Xu, X., Liu, C., Li, J., et al., 2022. Trajectory clustering for SVR-based Time of Arrival estimation. *Ocean Engineering*, *259*:111930. https://doi.org/10.1016/j.oceaneng.2022.111930

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

Xinlong, P., Wang, H., Cheng, X., et al., 2020. Online detection of anomaly behaviors based on multidimensional trajectories. *Information Fusion*, *58:*40–51. https://doi.org/10.1016/j.inffus.2019.12.009

Hong, Z., Chen, Y., Mahmassani, H. S., 2018. Recognizing network trip patterns using a Spatio-Temporal Vehicle Trajectory clustering algorithm. *IEEE Transactions on Intelligent Transportation Systems*, *19*(8):2548–2557. https://doi.org/10.1109/tits.2017.2754401

Yi, D., Chen, W., Liu, C., 2019. Trajectory clustering aided personalized driver intention prediction for intelligent vehicles. *IEEE Transactions on Industrial Informatics*, *15*(6):3693–3702. https://doi.org/10.1109/tii.2018.2890141

Advani, C., Chung, E., Boyle, L. N., 2022. Bi-level clustering of vehicle trajectories for path choice set and its nested structure identification. *Transportation Research Part C-emerging Technologies*, *144*:103895. https://doi.org/10.1016/j.trc.2022.103895

Hong, Z., Chen, Y., Mahmassani, H. S., et al., S. 2017. Commuter ride-sharing using topology-based vehicle trajectory clustering: Methodology, application and impact evaluation. *Transportation Research Part C-emerging Technologies*, *85:*573–590. https://doi.org/10.1016/j.trc.2017.10.020

Jin, F., Hua, W., Francia, M., et al., 2023. A survey and experimental study on Privacy-Preserving Trajectory Data Publishing. *IEEE Transactions on Knowledge and Data*

Unedited

Engineering, *35*(6):5577–5596. https://doi.org/10.1109/tkde.2022.3174204

Gaffney, S., Smyth, P., 1999. *Trajectory clustering with mixtures of regression models*. 63-72 https://doi.org/10.1145/312129.312198

Atev, S., Miller, G., Papanikolopoulos, N., 2010. Clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *11*(3):647–657. https://doi.org/10.1109/tits.2010.2048101

Benkert, M., Gudmundsson, J., Hübner, F., et al., 2008. Reporting flock patterns. *Computational Geometry: Theory and Applications*, *41*(3):111–125. https://doi.org/10.1016/j.comgeo.2007.10.003

Gariel, M., Srivastava, A. K., Feron, E., 2011. Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*, *12*(4):1511–1524. https://doi.org/10.1109/tits.2011.2160628

Chen, C., Ye, Z., Hu, F.,et al., 2021. Vehicle trajectory-clustering method based on road-network-sensitive features. *Journal of Intelligent & Fuzzy Systems*, *41*(1):2357–2375. https://doi.org/10.3233/jifs-211270

Yu, Q., Luo, Y., Chen, C.,et al., 2019. Trajectory similarity clustering based on multi-feature distance measurement. *Applied Intelligence*, *49*(6):2315–2338. https://doi.org/10.1007/s10489-018-1385-x

Besse, P., Guillouet, B., Loubes, J., et al., 2016. Review and Perspective for Distance-Based Clustering of Vehicle Trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *17*(11), 3306–3317. https://doi.org/10.1109/tits.2016.2547641

Rousseeuw, P. J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*:53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Davies, D. B., Bouldin, D. W., 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2):224–227. https://doi.org/10.1109/tpami.1979.4766909

Caliński, T., Harabasz, J., 1974. A dendrite method for cluster analysis. *Communications in Statistics*, *3*(1):1–27. https://doi.org/10.1080/03610927408827101

Wang, W., Xia, F., Nie, H., et al., 2021. Vehicle trajectory clustering based on dynamic representation learning of internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(6):3567–3576. https://doi.org/10.1109/tits.2020.2995856

Guo, M., Jin, X., Pissinou, N., et al., 2015. In-Network trajectory privacy preservation. *ACM Computing Surveys*, *48*(2):1–29. https://doi.org/10.1145/2818183

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

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

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

Ha, Wang., Zhengquan, Xu., 2018. Differential privacy preserving method for trajectory clustering. *Journal of Huazhong University of Science and Technology(Natural Science Edition)*, 46(1):32-36. (in Chinese) 10.13245/j.hust.180107

Ma, Z., Zhang, T., Liu, X., et al., 2019. Real-Time Privacy-Preserving data release over vehicle trajectory. *IEEE Transactions on Vehicular Technology*, *68*(8):8091–8102. https://doi.org/10.1109/tvt.2019.2924679

Arif, M., Chen, J., Wang, G.,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

Zhao, X., Pi, D., Chen, J., 2020. Novel trajectory privacy-preserving method based on clustering using differential privacy. *Expert Systems With Applications*, *149*:113241. https://doi.org/10.1016/j.eswa.2020.113241

Xiulin, Sun., lin, Lu., 2022. Privacy-preserved vehicular trajectory embedding federated learning and clustering. *Journal of Nanjing Normal University(Engineering and Technology Edition)*, 22(2):80-86. (in Chinese) 10.3969/j.issn.1672-1292.2022.02.012

Xie, X., Wu, B., Hou, B., 2023. BEPHAP: A Blockchain-based Efficient Privacy-Preserving Handover Authentication Protocol with key agreement for Internet of Vehicles. *Journal of Systems Architecture*, *138*:102869. https://doi.org/10.1016/j.sysarc.2023.102869

Lu, L., Lin, Y., Wen, Y., et al., 2023. Federated clustering for recognizing driving styles from private trajectories. *Engineering Applications of Artificial Intelligence*, *118*:105714. https://doi.org/10.1016/j.engappai.2022.105714

Cho, K., Van Merriënboer, B., Gulcehre, C., et al., (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* :1406.1078. https://doi.org/10.48550/arxiv.1406.1078

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, 1(14):281–297

Sculley, D., 2010. *Web-scale k-means clustering*. 1177-1178. https://doi.org/10.1145/1772690.1772862

Unedited

Park, H., Jun, C., 2009. A simple and fast algorithm for K-medoids clustering. *Expert Systems With Applications*, *36*(2):3336–3341. https://doi.org/10.1016/j.eswa.2008.01.039

Ng, Andrew Y., Jordan, Michael I., Weiss, Yair., 2001. On Spectral Clustering: Analysis and an Algorithm. *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, 849–856.

Kluger, Y., Basri, R., Chang, J. T., et al., 2003. Spectral Biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, *13*(4):703–716. https://doi.org/10.1101/gr.648603

Zhang, T., Ramakrishnan, R., Livny, M., 1996. BIRCH. *ACM Sigmod Record*, *25*(2):103–114. https://doi.org/10.1145/235968.233324

Dunn, J., 1973. A fuzzy relative of the ISODATA process and its use in detecting Compact Well-Separated clusters. *Journal of Cybernetics*, *3*(3):32–57. https://doi.org/10.1080/01969727308546046

Bezdek, J. C., 1981. Pattern Recognition With Fuzzy Objective Function Algorithms. *Plenum Press*.

Unedited